l'Ordinateur de poche

N° 23 - MAI-JUIN 1984

16 F

Quand on démonte l'imprimante du PC-1500

Canon contre mur de briques

A la tête d'un empire...

Cinq jeux pour le PB-100

Des idées, des programmes pour vos machines

Belgique 130 FB - Canada 1.95 $C - Suisse 5 FS

M 185 - 23 - 16 F

GIBRAT

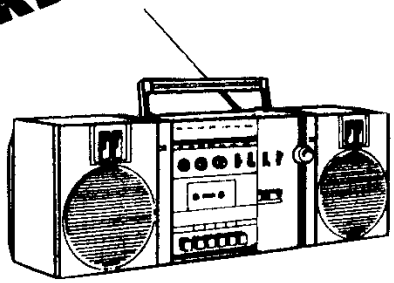
ÉCOUTEZ VOTRE ORDINATEUR INDIVIDUEL

L'ORDINATEUR INDIVIDUEL

ET

GILDA presse

MAINTENANT 2 FOIS PAR MOIS



vous proposent

LE PETIT ORDINATEUR ILLUSTRÉ

En MAI les n°s 9 et 10 de ce **magazine radio** seront émis dans la semaine du 1er au 7 et dans celle du 15 au 21 sur les antennes suivantes

Aix-en-Provence : RMP, 100 MHz
Mercredi, entre 19 h et 19 h 30

Belfort : Radio-Soleil, 88.1 MHz
Lundi, entre 13 h 05 et 13 h 35

Besançon : RVF, 98.1 MHz
Lundi, entre 17 h 30 et 18 h

Bordeaux : Radio 100, 94.3 MHz
Samedi, entre 19 h et 19 h 30

Clermont-Ferrand : MU, 96.2 MHz
Mardi, entre 19 h et 19 h 30

Dijon : Radio 2000, 90.7 MHz
Lundi, entre 19 h et 19 h 30

Gap : RTM, 90 MHz
Lundi, entre 18 h et 18 h 30

Grenoble : RTA, 90.7 MHz
Samedi, entre 13 h 30 et 14 h

Lannion : Pays de Trégor, 95.5 MHz
Samedi, lors de « Médiagora » de 19 h à 19 h 30

Le Mans : Le Mans FM, 104 MHz
Vendredi, entre 17 h 30 et 18 h

Lille : Radio Contact, 93.4 MHz
Jeudi, entre 20 h 30 et 21 h

Limoges : HPS, 102.7 MHz
Vendredi, entre 16 h 30 et 17 h

Lyon : Radio Bellevue, 94.9 MHz
Mercredi, entre 17 h 30 et 18 h

Montpellier : Radio Alligator, 94.5 MHz
Samedi, entre 9 h 30 et 10 h

Nancy : Rock'In Chair, 95.8 MHz
Tous les mardis, entre 24 h et 1 h du matin

Nantes : Atlantic FM, 96.8 MHz
Mercredi, entre 16 h 45 et 17 h 15

Orléans : Orléans FM, 93.6 MHz
Mercredi, entre 14 h et 14 h 30

Paris : Radio Gilda, 103.5 MHz
Jeudi, entre 19 h 30 et 20 h

Poitiers : RPO, 90 MHz
Jeudi, entre 18 h et 18 h 30

Rouen : VRL 104 MHz
Vendredi, entre 18 h 30 et 19 h

Rennes : RBS, 89.1 MHz
Samedi, entre 14 h et 14 h 30

Seine & Marne : Radio 77, 102.9 MHz
Mercredi, entre 20 h 30 et 21 h

Strasbourg : Nuée Bleue, 89.5 MHz
Vendredi, lors de « Fil en aiguille », de 20 h 30 à 22 h 30

Toulouse : Radio Occitania, 99.1 MHz
Samedi, entre 18 h 30 et 19 h

Tours : Méga-Tours, 103 MHz
Dimanche, entre 10 h 30 et 12 h

1
NOTRE COUVERTURE
Dernièrement, l'illustrateur Jean-Pierre Gibrat est venu jusqu'à notre rédaction en empruntant le métro. Un carnet, un crayon, quelques croquis. Comme on peut le voir, il songeait aux ordinateurs de poche !

5
ÉDITORIAL
... où vous apprendrez que l'équipe qui fait votre journal travaille dès maintenant sur un projet qui vous intéresse au premier chef.

12
A VOS CLAVIERS

15
MAGAZINE

18
GROSSE IMPRESSION
Quand on programme un PC-1211 pour imprimer des textes lisibles à plusieurs mètres de distance.

20
AUX COMMANDES D'UN PAYS IMAGINAIRE
Votre 702 P abrite tout un pays à qui vous devez procurer des emplois, de la nourriture, des écoles, etc. Par bonheur, il ne s'agit que d'une simulation.

23
OÙ EST LA SORTIE ?
Dans le dédale où vous errez, Ariane n'est pas là pour vous aider avec son fil. Mais vous pouvez aussi construire vous-même les plans du labyrinthe (programme pour PC-1251).

25
L'ADRESSAGE INDIRECT
Avec la plupart des Basics de poche, l'adressage indirect est un outil souple et facile à utiliser.

27
SUS AUX PSEUDO-PREMIERS !

Laurent Boulanikian, Christian Boyer, Gilles Bransbourg, Frank Broutin, Jacques Deconchat, Bernard Elman, François Fayard, Pierre Flener, Jérôme Gaudin, Pierre Ladislas Gedo, Luis Godinho, Enrico Induni, Pedro Inigo Yanez, Curt Knabe, Renée Koch, Jean-Christophe Krust, Xavier de La Tullaye, Jean-Christophe Laune, Raoul Lebastard, Jean-Charles Lemasson, Etienne Maetz, Thierry Mouton, Thierry Muller, Claude Nowakowski, Jean-Marc Odobez, Alain Pechmajou, Jean-Marc Pluéger, Jean-Pierre Raby, Jean-Jacques Santin, Robert Seraline, Lucien Strebler, Philippe Tenant, André Turlure.

l'Ordinateur de poche

16 F mai-juin 1984

n° 23

recherche, classement, etc... Douze options vous sont offertes.

34
LE PB-100 S'AMUSE
Cinq programmes de jeux pour le petit poquette de Casio.

38
UNE AUTOROUTE ET DES GRENOUILLES
Un classique du genre, adapté ici au ZX 81 : les grenouilles sont intrépides, aidez-les à ne pas se faire écraser !

Donnerez-vous votre langue au chat ?

42
UN CANON CONTRE UN MUR DE BRIQUES
Casser des briques, voilà le but du jeu ! Tout se déroule sur l'écran du X-07, mais l'ordinateur n'en souffrira pas...

44
LA LEÇON D'ANATOMIE
Voyage à l'intérieur de l'imprimante du PC-1500. La mise en pièces (détachées) est illustrée par un reportage photographique.

47
MISEZ P'TIT : OP'TIMISEZ
HP-41 C : le résultat des deux précédents défis. Les records sont encore tombés.

49
PARLEZ-VOUS LE MORSE ?
Le PC-1251, lui, le parle couramment, et il peut très bien vous aider à l'apprendre.

51
LES POINTS FORTS DU FORTH
Pour comprendre ce langage de programmation, il faut d'abord oublier tout ce que l'on connaît du Basic.

53
DES RÉSULTATS QUI PRÉSENTENT BIEN
Pourquoi faire compliqué quand il existe une solution simple ? Laissons le PC-1500 s'occuper des racines et des fractions...

54
CRÉATION DE TABLEAUX
Il ne s'agit pas de peinture, mais d'un tout autre art : la programmation. Tableaux à une ou deux dimensions, comment les établir et s'y retrouver.

59
AH ! SI VOUS AVIEZ SU...

Abonnements : Muriel Watremez assistée de Sylvie Trumel et de Cécilia Mollicone

l'Ordinateur de poche est une publication du

groupe tests publications

5 place du Colonel Fabien
75491 PARIS CEDEX 10
Téléphone : (1) 240 22 01
Télex : LORDI 215 105 F

Directeur de la publication : Jean-Luc Verhoye

intégrale, ou partielle, faite sans le consentement de l'auteur ou de ses ayants-droit ou ayants-cause est illicite » (alinéa 1er de l'Art. 40). Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contre-façon sanctionnée par les Art. 425 et suivants du Code Pénal.

OJD DIFFUSION · PUBLICITÉ 1982

BVP Bureau de Vérification de la Publicité

Notre publication contrôle les publicités commerciales avant insertion pour qu'elles soient parfaitement loyales. Elle suit les recommandations du Bureau de Vérification de la Publicité. Si, malgré ces précautions, vous aviez une remarque à faire, vous nous rendriez service en écrivant au BVP, BP 4508, 75362 PARIS CEDEX 08.

Vos poquettes à la une.

De la découverte

La découverte du PB-100
par Pierrick Moigneau
Série verte - Format : 14,5 x 21 - 168 pages
90,00 FF

Ce livre dévoile progressivement toutes les facettes de l'ordinateur de poche Casio PB-100 et du Basic à l'aide de nombreux exemples d'application, permettant ainsi aux novices d'entrer en douceur dans le monde de l'informatique.

La découverte du PC-1500
par Jean-Pierre Richard
Série verte - Format : 14,5 x 21 - 208 pages
100,00 FF

Un ouvrage pour les néophytes curieux qui veulent en savoir plus sur leur ordinateur de poche PC-1500 (ou TRS-80 PC-2). Quelles instructions et commandes emploie-t-il ? Quels types de variables utilise-t-il ? Comment la mémoire est-elle structurée ? Toutes ces questions et bien d'autres trouvent leur réponse dans ce premier tome de " La découverte du PC-1500 ". Un manuel riche de tous les éléments nécessaires à la programmation en Basic, largement complété d'exercices, d'exemples d'application et... d'un index.

La découverte du FX-702 P
par Jean-Pierre Richard
Série Verte - Format : 14,5 x 21 - 216 pages
100,00 FF

Instructions et commandes, variables et mémoires, fonctions périphériques, cet ouvrage fournit aux débutants tous les éléments de base nécessaires à la programmation en langage Basic de la Casio FX-702 P. Il est complété par de nombreux exemples et exercices d'application.

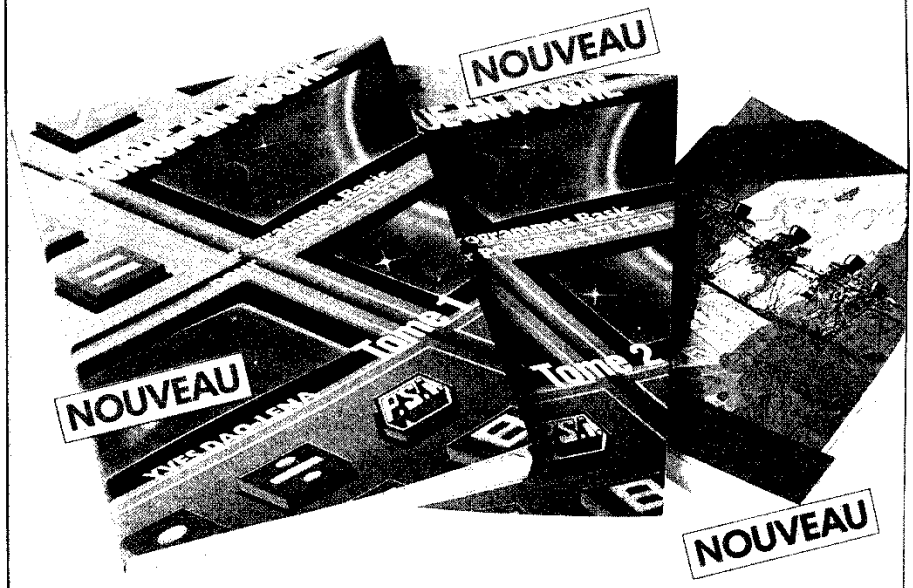
La découverte du PC-1251
par Jean-Pierre Richard - Série verte - 224 pages
100,00 FF

Comment aborder la programmation du Sharp PC-1251. C'est ce que découvriront les utilisateurs du PC-1251. Citons parmi les principaux thèmes traités : instructions et commandes, variables et mémoires, sous-programmes et traitement de chaînes, gestion des périphériques. Quelques exercices d'application et un index viennent compléter ce manuel qui fournira à l'utilisateur tous les éléments de base indispensables à la programmation en langage Basic.

La découverte de la TI-57
par Xavier de la Tilaye
Série verte - Format : 14,5 x 21 - 144 pages
80,00 FF

S'adressant aux débutants, cet ouvrage les conduira dans un langage clair, de l'élémentaire 2 + 2 à des programmes perfectionnés. Après une étude fonctionnelle de la calculette, la programmation est expliquée progressivement, de la conception à la réalisation en s'appuyant sur de nombreux exemples.

... aux programmes



Physique en poche - tome 1
30 programmes Basic pour PC-1500 et TI-55 II
par Yves Dao-Lena - 176 pages
110,00 FF

Physique en poche - tome 2
30 programmes Basic pour PC-1500 et TI-55 II
par Yves Dao-Lena - 176 pages
110,00 FF

60 programmes Basic, classés par thèmes, abondamment commentés permettent d'aborder certains des phénomènes physiques qui nous entourent : quelle est la distance parcourue par la lumière en un milliardième de seconde ? avec quelle force la Terre attirerait-elle un homme de masse de 70 kg, situé à un million de kilomètres de sa surface ?... Ecrits sur PC-1500 et TI-55 II les programmes de ces ouvrages pourront aisément être adaptés à toutes les machines grâce aux très nombreux commentaires. Chaque tome comprend 30 programmes.

Le PC-1251 à l'écran
20 programmes en Basic
par Pierrick Moigneau - 144 pages
90,00 FF

Pierrick Moigneau propose ici 20 programmes utiles ou ludiques, écrits en langage Basic, illustrant les possibilités de l'ordinateur de poche PC-1251 et de son extension CE-125. Des calculs d'horaires aux jeux d'astuces et de hasard, chaque programme est décrit, commenté et illustré d'organigrammes et d'exemples d'exécution. Utilisable également sur Tandy TRS-80 PC3.

Suites pour PC 1500
par Jean-François Sehan - Série bleue
Format 17 x 25 - 160 pages
90,00 FF

Destiné aux possesseurs de Sharp PC-1500 et de PC-2 Tandy, cet ouvrage aborde par l'exemple la programmation de ces " micros-poches ".

Alors, sur la base de ces 20 programmes, partez tout chuss sur les pistes enneigées, transformez votre " poquette " en piano, éditez les factures de vos clients et... inventez-en d'autres !...

Collèges - Poquettes et Maths
par Jacques Deconchat - Série verte
Format 17 x 25 - 200 pages **100,00 FF**

Ce livre destiné aux élèves des classes de collège et à tous ceux qui cherchent à mieux saisir les techniques de programmation des calculatrices programmables, propose 35 programmes d'arithmétique, d'algèbre et de géométrie.

Récréations pour TI-57 - Tome 1
par Jacques Deconchat - Série verte
Format 17 x 25 - 168 pages **90,00 FF**

Un recueil de quarante-cinq programmes de jeux très divers adaptés pour l'ordinateur de poche TI-57. Un exemple d'exécution est fourni avec chaque programme permettant de vérifier son bon fonctionnement et de mieux percevoir les différentes techniques d'affichage utilisées.

Récréation pour TI-57 - Tome 2
par Jacques Deconchat - Série bleue
Format 17 x 25 - 176 pages **90,00 FF**

45 nouvelles idées de jeux pour votre TI-57. Cependant des indications sur l'adaptation à d'autres machines sont fournies en annexe.

Vendredi, entre 17 h 30 et 18 h

Lille : Radio Contact, 93.4 MHz
Jeudi, entre 20 h 30 et 21 h

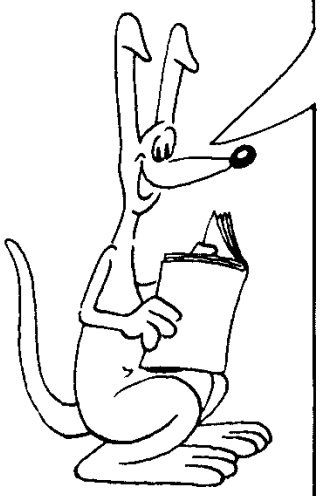
Limoges : HPS, 102.7 MHz
Vendredi, entre 16 h 30 et 17 h

Lyon : Radio Bellevue, 94.9 MHz
Mercredi, entre 17 h 30 et 18 h

Strasbourg : Nuée Bleue, 89.5 MHz
Vendredi, lors de « Fil en aiguille », de 20 h 30 à 22 h 30

Toulouse : Radio Occitania, 99.1 MHz
Samedi, entre 18 h 30 et 19 h

Tours : Méga-Tours, 103 MHz
Dimanche, entre 10 h 30 et 12 h



éditorial

D'Op en LIST

Depuis plus de trois ans, *l'Ordinateur de poche* publie régulièrement des articles ayant trait à la programmation. Astuces pour un matériel précis ou de portée plus générale, astuces développées sur plusieurs pages ou signalées en quelques lignes... Astuces, astuces, la programmation est tout entière faite d'astuces.

Parfois, c'est un jeu qui sert de prétexte pour expliquer comment tirer parti d'une instruction ou pour éclairer un algorithme. Mais, dans tout cela, **le plus fascinant des jeux,** le plus enrichissant, **c'est encore la programmation.**

L'équipe qui réalise *l'Op* a toujours aidé celles et ceux qui se passionnent pour la programmation. Jusqu'à présent, et mis à part le ZX 81, nous nous étions limités aux machines de poche. Nous avons décidé d'élargir notre horizon et de nous intéresser aussi aux ordinateurs un peu plus encombrants que l'on utilise chez soi parce qu'ils ne fonctionnent pas sans le réseau EDF. C'est vrai, ils ont un fil à la patte... En contrepartie, ils offrent un clavier permettant une frappe rapide, et un « grand écran ».

Evidemment, nous n'oublions pas l'informatique de poche. Depuis 1981, nous vous avons montré que nous avons un faible pour elle... Indépendamment de cela, nous sommes convaincus que la plupart d'entre vous seront curieux de découvrir les possibilités d'autres ordinateurs. Et vous savez comme nous qu'en matière de programmation, les idées à puiser ou à creuser ne dépendent pas nécessairement du matériel utilisé.

A l'avenir, ce n'est donc pas *l'Op* que vous trouverez chez votre marchand de journaux, mais LIST, et cela dès la fin juin. Nos abonnés le recevront à domicile quelques jours plus tôt. Notre équipe s'y emploie.

Depuis trois ans, beaucoup d'entre vous ont pris l'habitude de nous adresser par courrier leurs suggestions, leurs idées, leurs propositions d'articles. Notre adresse n'a pas changé. **Vous pouvez donc, dès à présent, écrire à LIST...**

☐ l'Op

RÉDACTION-RÉALISATION
Directeur de la rédaction : Bernard Savonet
Rédacteur en chef : Jean Baptiste Comiti
Rédaction : Anne-Sophie Dreyfus
Secrétaire de rédaction : Eliane Gueylard
Assistante de rédaction : Maryse Gros
Administration : Michelle Aubry

Ont collaboré à ce numéro : Thierry Akouka, Christian Allois, Michel Arditti, Xavier Benigni, Laurent Boulanikian, Christian Boyer, Gilles Bransbourg, Frank Broutin, Jacques Deconchat, Bernard Elman, François Fayard, Pierre Flener, Jérôme Gaudin, Pierre Ladislas Gedo, Luis Godinho, Enrico Induni, Pedro Inigo Yanez, Curt Knabe, Renée Koch, Jean-Christophe Krust, Xavier de La Tullaye, Jean-Christophe Laune, Raoul Lebastard, Jean-Charles Lemasson, Etienne Maetz, Thierry Mouton, Thierry Muller, Claude Nowakowski, Jean-Marc Odobez, Alain Pechmajou, Jean-Marc Pluéger, Jean-Pierre Raby, Jean-Jacques Santin, Robert Seraline, Lucien Strebler, Philippe Tenant, André Turlure.

Illustrations : Frapar, Jean-Pierre Gibrat, Bernard Helme, Hervé Marly, Alain Mirial, Nestor, Alain Prigent, Jean-Marc Rubio, Nicolas Spinga.

ÉDITION-PUBLICITÉ-PROMOTION
Éditeur : Jean-Pierre Nizard
Assistante d'édition : Maryse Marti
Publicité : Jean-Daniel Belfond

VENTES
Diffusion NMPP : Sophie Marnez
Abonnements : Muriel Watremez
assistée de Sylvie Trumel et de Cécilia Mollicone

l'Ordinateur de poche
est une publication du
groupe tests publications
5 place du Colonel Fabien
75491 PARIS CEDEX 10
Téléphone : (1) 240 22 01
Télex : LORDI 215 105 F

Directeur de la publication : Jean-Luc Verhoye

Tarif d'abonnement (10 numéros)
France : 130 FF TVA 4 % incluse, Belgique : 1150 FB, Suisse : 40 FS, Canada* : 17$C, autres pays* : 170 FF, Afrique francophone : 205 FF, autre Afrique, Amérique : 255 FF, Asie, Océanie : 295 FF.
* Tarif par avion

La loi du 11 mars 1957 n'autorisant, aux termes des alinéas 2 et 3 de l'Art. 41, d'une part que « les copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective », et, d'autre part, que les analyses et les courtes citations dans un but d'exemples et d'illustrations, « toute représentation ou reproduction intégrale, ou partielle, faite sans le consentement de l'auteur ou de ses ayants-droit ou ayants-cause est illicite » (alinéa 1er de l'Art. 40). Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contre-façon sanctionnée par les Art. 425 et suivants du Code Pénal.

OJD DIFFUSION · PUBLICITÉ 1982

BVP Bureau de Vérification de la Publicité

Notre publication contrôle les publicités commerciales avant insertion pour qu'elles soient parfaitement loyales. Elle suit les recommandations du Bureau de Vérification de la Publicité. Si, malgré ces précautions, vous aviez une remarque à faire, vous nous rendriez service en écrivant au BVP, BP 4508, 75362 PARIS CEDEX 08.

LE M
MULTI-C

Feldman, Calleux & Associés

ICRO
CARTES.

CANON X 07.
INVENTEZ VOTRE PROGRAMMATHEQUE.

NOMBREUX SONT LES ORDINATEURS QUI PARLENT. LE CANON X 07 EST L'UN DES RARES A VOUS ECOUTER. SA FORCE ? UNE PETITE CARTE EXTRAORDINAIRE A DOUBLE POUVOIR. POUR ETENDRE LES CAPACITES DU X 07, MAIS SURTOUT POUR REALISER ET CONSERVER VOS PROPRES PROGRAMMES COMME VOUS L'ENTENDEZ ET... A LA CARTE.

PRATIQUE, IL PARLE EN BASIC, LE LANGAGE ORDINATEUR FACILE A APPRENDRE.

AVEC SES NOMBREUSES CASSETTES ET CARTES LOGICIELLES A PROGRAMMES AUSSI ELABORES ET DIVERSIFIES QUE LA GESTION DE STOCKS, LA PRISE DE COMMANDES, L'ORGANISATION DE RENDEZ-VOUS..., CANON X 07 A EGALEMENT BIEN D'AUTRES ATOUTS.

GRACE A SES MULTIBRANCHEMENTS : MACHINE A ECRIRE, IMPRIMANTE, ORDINATEUR, MODEM... CE TOUT PETIT ORDINATEUR A TROUVE PLUS D'UN MOYEN POUR DEVENIR GRAND.

X 07, LE MICRO-ORDINATEUR VRAIMENT PERSONNEL. LE PREMIER MICRO-ORDINATEUR MULTI-CARTES.

pour une documentation détaillée

ır 10 métiers de l'informatique

DE L'ANNEE

, je désire recevoir gratuitement (et sans aucun engagement) une documentation détaillée la formation EDUCATEL d'enseignement personnalisé des 10 métiers informatiques.
ırouverai pour chaque métier préparé le plan de formation complet, son niveau d'accès, rogramme des travaux pratiques, sa durée et son prix.
ı le désire, une orientation et des conseils personnels me seront fournis gratuitement.
ıeux également téléphoner à EDUCATEL au (1) 208.50.02 (demander Madame LAMY).

OP 0006

M _____ Prénom _____
esse _____
de postal _____ Ville _____
éphone (facultatif) _____ Age _____
vaillez-vous ? OUI ☐ NON ☐ Niveau d'études _____
écisez le métier qui vous intéresse :

DUCATEL G.I.E. Unieco Formation
00 X - 76025 ROUEN CEDEX
ır Canada, Suisse, Belgique : 49, rue des Augustins - 4000 Liège
ır TOM-DOM et Afrique : documentation spéciale par avion.

ou téléphonez à Paris (demandez Madame LAMY)
(1) 208.50.02

POUR MIEUX CHOISIR VOTRE ORDINATEUR ET POUR MIEUX L'UTILISER :

Vous y trouverez :
l'actualité et les tendances de l'informatique individuelle
- les bancs d'essais des principaux matériels
- des panoramas et des tests comparatifs
- le point des grandes manifestations internationales
- des articles d'initiation
- des synthèses
- des programmes
- des interviews "exemplaires"
- des conseils
- des idées
- des astuces

L'ORDINATEUR INDIVIDUEL

18 ORDINATEURS : D'UN BASIC A L'AUTRE
Non aux langages
Choisir votre bibliothèque CP/M
Programmer facilement en assemblage

LA COTE DE L'OCCASION

SOUS LE SIGNE DES JEUX
68 logiciels testés sur 10 ordinateurs : les étoiles de L'OI
Apprendre : est-ce un jeu ?
La création d'un jeu

PARLEZ-VOUS SHADOCK ? (sur ZX 81)

PROGRAMMES, TRUCS ET ASTUCES POUR :
TI 99/4A, ZX 81, Vic 20
Oric 1, Apple 2, Atom
TO 7, TRS 80, Dai, BBC

ESSAIS :

L'IBM PC JUNIOR

L'ORDINATEUR INDIVIDUEL

QUELS ORDINATEURS DEMAIN ?
Souris, écran tactile, crayon lumineux : le clavier sur la touche ?
Les écrans à fenêtres
Les logiciels intégrés
Les composants du futur

ESSAIS : IBM 186, HP 150, Lisa, Aquarius, MS Win, Executive 1, etc.

PROGRAMMES ET ASTUCES :
Apple 2, CBM 64
Oric, ZX Spectrum
ZX 81, HP 75 C
TI 99/4A, etc.

L'ORDINATEUR INDIVIDUEL

LE GUIDE DES PORTABLES : 85 ORDINATEURS

ESSAIS : Casio FP 200, Atari 800 XL, MPF 2, TRS 80 modèle 100, Tandy, Tytus, Tulex

PROGRAMMES, TRUCS ET ASTUCES POUR :
TI 99/4A, ZX 81, Oric, Atari, MPF 2, SR 41C, TI 57, RHC 2, CBM, Dragon 32, New Brain, Apple 2, ZX/81, etc.

Feldman, Calleux & Associés

L'ORDINATEUR INDIVIDUEL
La Référence

SPECIAL CARRIERES INFORMATIQUES

10 métiers informatiques

- Analyste programmeur
- Analyste
- Programmeur d'application
- Programmeur sur micro-ordinateur
- Pupitreur
- Opérateur sur ordinateur
- Opératrice de saisie
- Spécialisation en langage informatique
- Correspondant informatique
- B.T.S. Services informatiques

5 réponses à vos questions

L'INFORMATIQUE, QUEL AVENIR ?

Devenir informaticien en 1984, c'est choisir une carrière d'avenir, avec l'assurance de trouver immédiatement de nombreux débouchés, et des perspectives d'autant plus intéressantes que la place de l'ordinateur ne cesse de s'accroître dans tous les domaines : économique, social, administratif, etc.

En 1984, plus de 300.000 personnes travaillent directement dans l'informatique, les places sont donc nombreuses tant pour les femmes que pour les hommes, et ceci à tous les échelons de la hiérarchie.

Les chiffres de l'A.N.P.E. le prouvent : actuellement, plus de la moitié des postes proposés par les employeurs à des informaticiens (programmeur, analyste programmeur, etc.) ne sont pas pourvus, faute de candidats en nombre suffisant.

COMMENT DEVENIR INFORMATICIEN ?

En suivant une formation qui associe un enseignement théorique complet, régulièrement remis à jour, à un enseignement pratique : exercice sur micro-ordinateur ou stages sur matériel IBM.

Notre objectif : vous rendre opérationnel pour que vous puissiez aborder dans les meilleures conditions les réalités de la vie professionnelle.

QUE FAUT-IL POUR REUSSIR UNE FORMATION EN INFORMATIQUE ?

L'informatique, ce n'est pas compliqué ! Quel que soit votre niveau de formation, vous pouvez apprendre en quelques mois, par les moyens les plus modernes, le métier de l'informatique qui vous convient le mieux. Pour vous y aider, nous faisons appel à des professeurs, spécialistes de l'informatique qui, par leur expérience professionnelle, vous permettront de mieux vous préparer au monde du travail.

COMMENT APPRENDRE « PRATIQUEMENT » L'INFORMATIQUE ?

Vous savez combien il est important aujourd'hui d'être opérationnel lorsque l'on cherche un emploi, ou que l'on désire changer de métier, surtout dans un secteur de pointe tel que celui de l'informatique.

Si vous le souhaitez, vous participerez à l'un des stages pratiques que nous organisons dans notre Centre de Formation.

Pédagogues mais aussi professionnels de l'informatique, nos professeurs vous ferons travailler sur le matériel le plus utilisé dans les entreprises équipées (ordinateur IBM 34), et vous pourrez être ainsi confronté aux situations que l'on rencontre quotidiennement dans une entreprise ou un service informatique.

QUE SE PASSE-T-IL A LA FIN DE LA FORMATION ?

Pour compléter votre formation, vous pouvez, à la fin de votre étude, effectuer un stage en entreprise. Nous nous chargerons de contacter des entreprises afin de vous trouver un terrain de stage.

Si vous le souhaitez, nous soutiendrons également votre candidature auprès des employeurs lorsque vous chercherez un emploi.

Depuis 10 ans, EDUCATEL prépare aux carrières de l'informatique. Chaque année, nous formons 5.000 informaticiens, depuis l'opératrice de saisie jusqu'à l'analyste.

SOGEX

Si vous êtes salarié, votre étude peut être prise en charge par votre employeur (loi du 16-7-1971 sur la formation continue).

EDUCATEL - 1083, route de Neufchâtel 3000X - 76025 ROUEN Cédex

G.I.E. Unieco Formation
Groupement d'écoles spécialisées.
Etablissement privé d'enseignement par correspondance soumis au contrôle pédagogique de l'Etat.

POSSIBILITE DE COMMENCER VOS ETUDES A TOUT MOMENT DE L'ANNEE

BON pour une documentation détaillée sur 10 métiers de l'informatique

OUI, je désire recevoir gratuitement (et sans aucun engagement) une documentation détaillée sur la formation EDUCATEL d'enseignement personnalisé des 10 métiers informatiques.

J'y trouverai pour chaque métier préparé le plan de formation complet, son niveau d'accès, le programme des travaux pratiques, sa durée et son prix.

Si je le désire, une orientation et des conseils personnels me seront fournis gratuitement.

Je peux également téléphoner à EDUCATEL au (1) 208.50.02 (demander Madame LAMY).

OP 0006

NOM _____ Prénom _____

Adresse _____

Code postal _____ Ville _____

Téléphone (facultatif) _____ Age _____

Travaillez-vous ? OUI ☐ NON ☐ Niveau d'études _____

Précisez le métier qui vous intéresse :

EDUCATEL G.I.E. Unieco Formation
3000X - 76025 ROUEN CEDEX

Pour Canada, Suisse, Belgique : 49, rue des Augustins - 4000 Liège
Pour TOM-DOM et Afrique : documentation spéciale par avion.

ou téléphonez à Paris (demandez Madame LAMY) (1) 208.50.02

Le guide des applications professionnelles

N° 5

chez votre marchand de journaux

LA GAMME DES LOGICIELS POCKET SOFT®

POUR SHARP PC-1500®

Distribution **XLOG**

PC-CALC 3 : Feuille électronique de calcul
PC-PLOT 3 : Utilitaire graphique de PC-CALC 3
PC-GRAPH : Représentation graphique
PC-WORD : Traitement de texte
PC-MACRO : Macro-assembleur
PC-HEX : Moniteur hexadécimal avec TRACE
PC-MATH : 15 fonctions mathématiques, statistiques et financières
PC-UTIL 2 : 19 instructions BASIC supplémentaires
PC-BANK : Gestion simultanée de 5 comptes bancaires
PC-PERT : Méthode PERT et chemin critique
PC-PLAN : Etablissement de plannings et échéancier
PC-POLYNOMIAL : Traitement des polynômes
PC-BUDGET : Gestion budgétaire

POUR CANON X-07®

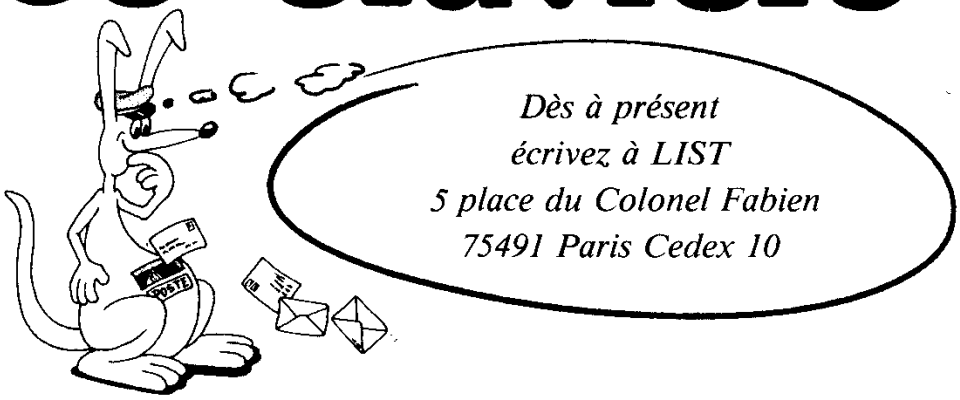
Distribution **CANON**

Logiciels pouvant échanger leurs données entre eux :
X-07 CALC : Feuille électronique de calcul avec graphiques
X-07 STAT : Statistiques avec graphiques
X-07 GRAPH : Représentation graphique
X-07 DATABASE : Base de données
X-07 INVOICE : Facturation
X-07 INVENTORY : Gestion des stocks

Autres logiciels :
X-07 TEXT : Traitement de texte.

Disponibles dans les boutiques micro-informatique

A vos claviers



Apostropher le FX-702 P

Le FX-702 P ne possède pas d'apostrophe. J'ai donc essayé de la remplacer par des guillemets : $ = "L"OP" ou PRINT "L"OP". J'obtiens une erreur : ERR-2. Maintenant, en mode programme, si je rentre la ligne : 1 INP $: PRT ":" ; $ et que j'assigne à $ un mot contenant des guillemets (L"OP, par exemple), le FX-702 P l'acceptera.

C'est un petit truc qui peut toujours servir.

Alexandre Odaux
59 Lomme

■ *Bravo pour le petit truc, il pourra servir, en effet.*

TI-58 en mer

Navigateur de plaisance, j'ai été très intéressé par les programmes proposés par Lucien Strebler.

Possesseur d'une TI-58 C, j'ai pu les utiliser, ou du moins les essayer, car leur utilisation en mer soulève un problème que je n'ai pu encore résoudre : l'alimentation de la TI.

Comme l'autonomie de la batterie rechargeable ne dépasse guère les 3 heures et que je n'ai pas de groupe électrogène à bord, il m'est difficile de l'utiliser même au cours d'une navigation de courte durée.

La question que je soumets donc à Lucien Strebler est : existe-t-il un dispositif simple qui permette d'alimenter une TI-58 ou 59 à partir d'une batterie de 12 volts ?

Avec mes remerciements.

Jacques Popineau
14 Caen

■ *L'alimentation d'une TI-58 ou d'une autre calculatrice similaire à bord d'un bateau ne pose aucun problème.*

Il suffit d'utiliser un convertisseur pour rasoir électrique. Ces appareils font du 220 V alternatif à partir du 12 volts continu de la batterie de bord.

Ils ont, pour la plupart, tendance à chauffer en utilisation prolongée. Pour éviter cela, il n'y a qu'à mettre la calculatrice en charge sans dépasser un quart d'heure d'affilée.

Pour ma part, j'ai utilisé une TI-59 de cette façon en la mettant en charge un quart d'heure par jour et cela pendant une croisière de trois semaines.

A noter que cet appareil permet aussi d'utiliser un ZX Spectrum. Dans ce cas, il n'est plus question de recharger, mais d'utiliser directement. Bien entendu, le téléviseur associé doit marcher sur 12 volts.

LS

Du pareil au même

J'ai remarqué dans certaines listes de programmes pour PB-100, notamment les programmes qui emploient une « imitation » de la fonction WAIT, une interversion de ce type : 100 FOR I = 1 TO 200 : PRINT : NEXT I au lieu de 100 FOR I = 1 TO 200 : NEXT I : PRINT. D'où ces erreurs peuvent-elles provenir ?

Amicalement.

Michel Guitton
13 Saint Chamas

■ *En fait, il ne s'agit pas d'erreurs. Avec un PB-100, l'instruction PRINT placée avant ou après NEXT I ne change rien à la pause.*

Misez p'tit... mais pas trop

Bravo pour votre rubrique « Misez p'tit » destinée aux utilisateurs de HP-41 C. Mais la frénésie de la réduction ne doit pas faire perdre la tête. En effet, je relève, page 24 de *l'Op* 20, au sujet du programme de Jean-Claude Guérout une inexactitude qui mérite d'être signalée. Sur le programme lui-même, je ne trouve rien à redire.

Cette inexactitude concerne les économies réalisées en évitant l'emploi d'ENTER dans un programme entre deux nombres. Par exemple 1 ENTER 2 et 1 (SST) 2 font la même chose mais bien que le SST ne soit pas programmé, on ne gagne pas un seul octet !

Dans le second exemple, si ENTER n'est pas programmé, il l'est tout de

Hé ! Pas si vite ! Un peu de patience : LIST paraîtra fin juin.

même *de facto* puisque la 41 C insère systématiquement un octet nul entre deux nombres.

Robert Pulluard
Bleiswuk, Pays-Bas

■ *Certes, certes, je me suis trompé, mais... (écrit l'auteur un peu honteux de son erreur gentiment critiquée), mais il n'en reste pas moins que 1 (SST) 2 vaut mieux que 1 ENTER 2.*

D'une part, si l'on ne gagne pas un **octet** *comme trop vite écrit, on gagne bien un* **pas** *de programme. C'est un critère d'optimisation.*

Et d'autre part, si l'on mesure le temps d'exécution de 1 ENTER 2, on trouve 1 272 millièmes de seconde contre 1 252 millièmes seulement pour 1 (SST) 2. Gain : 20 millièmes de seconde.

N'oublions pas que la règle de ce jeu est d'optimiser, même — et surtout — les bouts de chandelles... Rien que pour le plaisir !

JCK

Enregistrer sur PB-100...

Avec mon PB-100 et son interface-cassette FA-3, j'ai remarqué que la durée d'enregistrement d'un programme variait suivant l'instruction utilisée.

Avec l'instruction SAVE, l'enregistrement d'un programme de 544 pas dure 1 minute et 26 secondes. Il ne dure plus que 35 secondes avec l'instruction SAVE A.

Quant au nom de programme, facultatif, il n'a aucune influence sur le temps d'enregistrement.

Alain Berne
42 Lorette

... et sur FX-702 P

Pour sauvegarder un programme tenant sur une seule zone, j'utilisais l'instruction SAVE. Et puis, j'ai essayé l'instruction SAVE ALL.

Alors j'ai pu observer que, pour des programmes courts, les temps d'enregistrement par SAVE et par SAVE ALL sont assez proches mais, pour des programmes longs, ils sont très différents.

L'instruction SAVE ALL permet un gain de temps appréciable.

Un gain de temps mais un gain de place aussi : la bande magnétique étant moins encombrée, elle pourra contenir un plus grand nombre de programmes, si tous son enregistrés par SAVE ALL.

Pierre-Marie Lebon
71 Chagny

A vos claviers

Encore plus court

La suite de Fibonacci, présentée dans *l'Op* 21 page 38, peut se retrouver en quatre petits pas dans la TI-57 normale : x ⇌ t SUM 7 2nd Pause RST. Il suffit alors de taper 1 R/S et les nombres de Fibonacci défilent.

Gilles Nicolas
Medea Algérie

Quatre pas suffisent pour obtenir la suite de Fibonacci sur TI-57 (LCD ou non) : x ⇌ t + 2nd Pause RST.

On initialise ainsi : 2nd C.t 1 RST R/S. Et les nombres s'affichent les uns après les autres.

Kerkeni Othman
69 Villeurbanne

■ *Dans le numéro 22, un programme en cinq pas donnait les mêmes résultats. On en est arrivé maintenant à quatre petits pas, et ce record ne sera sans doute jamais battu.*

Méli-mélo sur FX-702 P

Si vous n'avez pas peur de « planter » votre FX-702 P, voici une petite bizarrerie.

Un programme est dans la machine et vous voulez le protéger par un mot de passe.

Frappez alors PASS « mot de passe ». Puis très vite, presque simultanément, tapez EXE et éteignez la machine.

En la rallumant et en listant le programme qui était à l'intérieur, bien des choses auront changé (surtout à la fin).

Patrick Waechter
67 Bischwiller

■ *Une seule solution pour sortir le FX-702 P de là : CLR ALL !*

Le supérieur et l'inférieur

En matière de programmation, il est souvent utile de trouver entre deux nombres le plus grand et le plus petit. Il existe beaucoup d'algorithmes mais il en est deux (en fait le même) qui sont vraiment utiles et facilement programmables sur les PC-1500, 1251, 1211....

Soit à comparer deux nombres x et y. La variable Z doit contenir le plus grand des deux : Sup (x, y). On peut programmer :

IF X ≥ Y LET Z = X
IF Y > X LET Z = Y

Sur certaines machines, on pourra alléger ce test en : Z = Y : IF X ≥ Y LET Z = X ce qui revient au même en fin de compte.

Une troisième formule permet de calculer Sup (x, y) « en force » : Z=X* (X≥Y) + Y*(X<Y). Ici les expressions logiques (X≥Y) et (X<Y) valent 1 si *vrai* et 0 si *faux*. L'une et l'autre ne peuvent d'évidence être vraies en même temps. Alors, si X≥Y, l'expression vaut : Z=X*(1)+Y*(0) soit X. Et, si X<Y, elle vaut Z=X*(0) + Y*(1) soit Y.

Pour déterminer le plus petit des deux nombres, Inf (x, y), il suffit d'inverser les tests en (X≤Y) et (X>Y). Notez que si x=y alors Inf (x, y) = Sup (x, y) = x = y.

Enfin, on connaît une formule mathématique pour *calculer* (sans test informatique) Sup (x, y) ou Inf (x, y) :

Sup (x, y) = 1/2 (x + y + |x − y|) et Inf (x, y) = 1/2 (x + y − |x − y|), où, bien sûr, l'expression |x − y| se traduit en Basic par ABS (X − Y).

Nous obtenons donc enfin un algorithme unique calculant à volonté soit Sup (x, y) soit Inf (x, y) :

T = 1 ou −1 pour avoir Sup (x, y) ou Inf (x, y)
X = nombre x
Y = nombre y
Z = (X + Y + T * ABS (X − Y))/2.

Voilà, tout est dit.

Arnaud Beauregard
78 Bougival

Des drapeaux cousus main

Certains ordinateurs de poche ont des drapeaux, comme la HP-41 C ou les TI-58 et 59. Mais qu'en est-il du FX-702 P ?

Christian Mazzolin
57 Marly

■ *Le 702 P est dépourvu de tout drapeau. Cela dit, il est facile d'en créer soi-même, en les programmant. On utilisera une variable, disons A, qui prendra, selon les résultats obtenus pendant le déroulement du programme, la valeur 0 ou 1. En testant ensuite la valeur de cette variable, on pourra aiguiller le programme entre deux traitements différents en fonction de ce qui s'est produit plus tôt.*

Index des annonceurs

Canon ..... **p. 6 et 7**
Duriez ..... **p. 66**
Educatel ..... **p. 9**
L'Ordinateur Individuel ..... **p. 8**
L'Ordinateur Personnel ..... **p. 10**
Maubert Electronic ..... **p. 48**
Ordi 5 ..... **p. 67**
Pocket Soft ..... **p. 11**
POI ..... **p. 2**
PSI Diffusion ..... **p. 4**
Service Musique Equipement ..... **p. 14**
Vidéo Technologie ..... **p. 68**
Votre Ordinateur ..... **p. 65**

A vos claviers

Et puis un jour...

Vous êtes des dizaines de milliers à avoir découvert l'informatique de poche. Au début, bien évidemment, c'est toujours la même chose : on n'y connaît rien. Et puis un jour... Chacune de ces expériences a quelque chose d'original. Voici le récit qu'en fait l'un de nos lecteurs.

■ Comment se fait-il qu'un individu soit un jour attiré par les chiffres ? Mystère. Quand je suis entré au lycée, j'ai été très étonné par notre prof de physique. Il maniait sa règle à calcul avec une telle dextérité que l'on se demandait si ce n'était pas un peu truqué. Puis j'ai appris à me servir de cet instrument. J'ai même compris pourquoi ça marchait. Les logs, la trigo, tout était bon pour jouer avec cette merveilleuse petite machine.

Texas en classe

En abordant la terminale, j'ai dû m'habituer à ce cauchemar qu'est le traitement des fonctions : y a-t-il des zéros, des points d'inflexion, des asymptotes ?

La règle à calcul était tombée un peu dans l'oubli, d'autant plus qu'elle n'était pas admise aux examens. A l'époque, on utilisait des tables de log à cinq décimales. Manque de chance, pour divers travaux de physique, il me fallait une décimale de plus. Cela m'a poussé vers le nouvel ordinateur de table HP9820 que l'école venait de s'offrir.

Quelle découverte ! Il y avait même un traceur de courbes. Inutile de dire que, dès ce jour, j'ai pressenti que les calculs répétitifs seraient du ressort de l'ordinateur. Et me voilà parti : régression linéaire, zéro de fonctions, courbes diverses, trajets de rayons lumineux dans des combinaisons de lentilles, et j'en passe...

C'est à ce moment que je me suis offert une Texas SR-50. Miracle de la technique moderne : une mémoire, fonctions trigonométriques, hyperboliques, l'ENIAC dans ma poche ! Cette machine m'a permis de préparer mon Bac. Par la suite, je l'ai donnée à mon père qui l'utilise toujours.

En arrivant à l'École Polytechnique de Lausanne, je me suis résolu à acheter une calculatrice autrement plus puissante : la TI-58. J'allais enfin pouvoir travailler correctement. Et de fait, pendant plusieurs années, cet ordinateur de poche allait me rendre d'énormes services. Pour l'anecdote, un de mes camarades possédait une TI-59 et nous avions concocté un admirable « Master Mind » (sur carte) pour les cours barbants...

Dans cette école, je me suis familiarisé avec les grosses « bécanes » (CDC, VAX,...) et certains langages évolués (Pascal, Fortran, Basic). Je me suis alors rendu compte de la jungle qui règne en matière de standards pour ce type de programmation. Comment réagir ? On me propose de suivre un cours sur l'assembleur du Z 80. Et vive l'assembleur ! Mais là aussi, à chaque processeur son langage. C'est encore la tour de Babel.

Les HP-41 C pointent alors le bout de leur nez. Cela me séduit, mais le nouveau PC-1211 me fait hésiter, hésiter... Et ma TI-58 me paraît de plus en plus petite.

Un OP pour mieux entendre

Une fois mes études terminées, je me lance dans une profession entièrement axée sur l'acoustique : haut-parleurs, enceintes acoustiques, acoustique interne des salles, sonorisations, sans parler de tous les problèmes courants de l'ingénieur électricien : circuits, réductions de réseaux... Décidément, cette TI-58 ne me suffit plus. D'autant que je lis une revue spécialisée (*l'Op* pour ne rien vous cacher) qui me fait des « Clins d'œil ». FX-702 P ? J'hésite encore. Et puis non ! Finalement, ce sera le PC-1500 à cause de sa table traçante.

Et me voilà bientôt propriétaire d'une jolie bibliothèque de programmes. Le tour est joué. Les plus fastidieux des calculs d'acoustique sont devenus dérisoires. La solution tient sur quelques bandes magnétiques. Il me suffit de charger les programmes utiles dans ma machine en fonction de mon agenda (avec 8 Ko de mémoire supplémentaire, vive « MERGE » !).

Ce dont je suis le plus fier pour l'instant, c'est un programme résolvant diverses équations d'acoustique interne de salle. Il fonctionne par menu avec touche « HELP » et commandes par défaut. Mais, les programmes écrits en Basic étant un peu lents à mon goût, je passe maintenant à la vitesse supérieure : le langage-machine.

L'âme d'un musicien

Le PC-1500, mon dernier OP ? Il a bien sûr l'interface RS-232, mais que penser du CC-40, du X-07, et des autres ? Pour moi, un ordinateur de poche doit être de poche et le PC-1500 (sans interface) me convient. S'il me fallait une mallette pour le transporter, il perdrait à mes yeux une grande partie de son intérêt. Autant travailler avec une machine de table. Cela me fait dire que les ordinateurs de poche sont, grâce à leur format, irremplaçables.

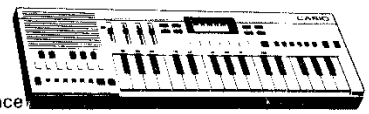
Mon PC-1500 devrait me satisfaire quelques années encore avant que, pris par une certaine démangeaison, je ne me décide à le reléguer au second plan au profit d'un OP plus puissant, plus rapide, et moins cher.

En attendant, quand j'ai besoin de me relaxer, j'écoute l'enregistrement d'un requiem ou d'une symphonie. Debussy, Mendelssohn, Verdi, Bach... Tout s'organise de façon claire autour d'un thème. Quelle perfection, quelle poésie ! Mais à propos, un programme que l'on développe, que l'on met au point, que l'on bichonne... N'y a-t-il pas là aussi un « je ne sais quoi » qui ressemble un peu à l'activité du compositeur de musique ? Je me sens tout à coup l'âme d'un Mozart de la petite informatique ! Vite, au travail, avant que l'inspiration ne me quitte...

☐ Enrico Induni

JOUEZ RAPIDEMENT DU CLAVIER !
c'est facile avec la ***TABLATURE ACOUSTIQUE***

la seule notation musicale à géométrie informatisée, utilisable directement par l'œil



et tous les **CASIO**
Exemple : PT-30,
1.150 F T.T.C.
franco de port en France

+ livret exclusif gratuit (14 transcriptions)

SERVICE MUSIQUE EQUIPEMENT, 38 r. Quincampoix 75004 PARIS

Bon pour le nouveau catalogue **CASIO** en couleurs + doc. complète
NOM : Prénom :
Adresse :
Code postal : Ville :

Magazine

■ UN LIVRE

Jeux en Basic sur ZX 81
Mark Charlton
Editions Sybex
Paris, 1983
Broché, 88 pages
Prix : 49 FF

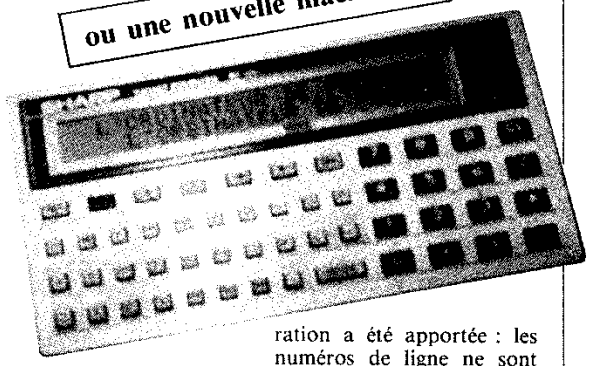
■ Une présentation avenante à première vue : un visage souriant sur la couverture et, au dos du livre, des photos d'écran qui correspondent aux programmes de jeux que l'on découvrira à l'intérieur. Car c'est bien d'un livre de programmes qu'il s'agit : 18 programmes, tous traduits en français, mais pratiquement pas commentés. Ils sont seulement présentés par quelques lignes de texte.

Ces programmes, inspirés pour la plupart des jeux dits « d'arcades », ont une forme assez élégante, bien adaptée au ZX 81 qui n'est pas particulièrement un champion sur le plan graphique.

On pourra regretter le côté un peu austère de la formule retenue ; seuls quelques dessins, pas très variés d'ailleurs (on retrouve seize fois le même...), viendront distraire le lecteur. Inutile de rechercher la moindre indication sur les astuces utilisées : ce livre est une suite de bons programmes, ni plus, ni moins.

□ JD

Le Sharp PC-1261 : un 1251 amélioré ou une nouvelle machine ?



■ Au premier coup d'œil, on croit se trouver devant un PC-1251 : même format « calepin », même clavier, même interrupteur 4 positions (OFF, RUN, PRO, RSV), même connecteur a 11 broches.

Mais il faut aussitôt se raviser : l'afficheur comporte non pas une seule ligne, mais deux lignes de 24 caractères.

Bien sûr les caractères deviennent petits, à peine 3,5 mm de hauteur, mais ils sont encore très lisibles et la molette de réglage du contraste est toujours présente. Pour pouvoir loger ces deux lignes, les témoins tels que ceux des modes angulaires ne sont plus affichés « en clair » sur l'écran mais écrits sur le boîtier, des traits s'allumant tout simplement en face.

Il y a encore quatre témoins en toutes lettres, mais positionnés verticalement à la gauche de l'afficheur : DEF, SHIFT, SMALL et カ ナ (Katakana). Nous n'avons pu obtenir ce dernier mode que par hasard lors d'un « reset », les touches alphabétiques donnant alors de curieux caractères japonais : un nouveau reset, et tout est rentré dans l'ordre.

Un exemple de l'utilité de ces deux lignes : en mode RUN, le calcul demandé et le résultat peuvent être présents simultanément à l'écran, commode !

La mémoire vive a été, comme sur le PC-1255 (voir *l'Op* 22, page 17), largement augmentée. A l'ordre MEM, le nouveau Sharp répond 9342 au lieu des 3486 octets du PC-1251.

Un examen attentif du clavier du PC-1261 montre finalement quelques différences avec celui du 51. Alors qu'en première fonction toutes les touches sont identiques, nous notons en Shift la modification de l'emplacement des parenthèses, l'absence du signe spécial E de l'exposant (le 1261 n'utilisant plus qu'un E normal), et surtout l'apparition de SML et HELP.

La première fonction permet d'obtenir l'alphabet non pas en lettres capitales, mais en minuscules. Et la seconde permet de lister tous les ordres BASIC du Sharp : en se positionnant à l'aide des quatre curseurs sur l'ordre désiré et en appuyant sur ENTER, on obtient un exemple d'emploi. Ainsi un *help* de l'ordre STEP fait afficher l'exemple succinct mais significatif suivant :

```
FOR L = 10 TO 0 STEP – 2
NEXT L
```

Cette facilité est agréable quand on a un « trou de mémoire » : bien plus rapide que d'aller chercher dans le manuel !

Au chapitre de la programmation, le BASIC est assez voisin de celui du PC-1251. Toutefois une amélioration a été apportée : les numéros de ligne ne sont plus limités à 999, mais à 65279 comme pour le PC-1500, et l'argument de STEP n'est pas nécessairement une valeur entière.

Un mieux aussi concerne la vitesse d'exécution : une boucle FOR I = 1 TO 1000 : NEXT I est exécutée en 7 s. (au lieu de 42 s.). Mais ce gain de vitesse vaut surtout pour les boucles vides. En effet FOR I = 1 TO 100 : A = SIN I : NEXT I est exécutée en 19 s. sur le 1261 contre 35 s. sur le 1251. On peut donc estimer qu'en moyenne la rapidité est augmentée d'un facteur compris entre 2 et 3.

En mode PRO sont stockés deux programmes indépendants : bien entendu le programme BASIC, mais aussi une sorte de petit tableur. C'est une surprise que ce tableur incorporé dont la programmation est assez simple. Supposons que nous ayons souvent à calculer des prix TTC à partir du prix hors taxes, la TVA étant à 18,6 %. Il suffit de rentrer en mode PRO la ligne :

```
#TVA = .186*HT:TTC = HT+TVA
```

De retour en mode RUN, l'ordre # liste les titres des formules entrées dans le tableur. Dans la liste devra donc se trouver #TVA. En

Magazine

se positionnant comme avec HELP sur la formule désirée et en appuyant sur ENTER, il s'affiche sur la première ligne
HT : TVA : TTC
Il est alors possible de rentrer sur la deuxième ligne le prix HT du produit, et s'afficheront automatiquement les résultats du calcul positionnés sous les titres TVA et TTC. Un nouveau calcul peut être repris en appuyant sur ENTER directement après l'affichage des résultats, toute autre touche faisant sortir du tableur.

Si l'on a plus de trois paramètres ou résultats, l'afficheur se comporte comme une fenêtre se décalant au fur et à mesure automatiquement vers la droite dans le tableau de chiffres. Le déplacement de cette fenêtre peut être forcé par l'emploi du curseur à droite ou à gauche.

Les différentes formules de tableur sont listées en mode PRO par l'ordre LIST#. La mémoire vive allouée au tableur est de 128 octets, mais on peut configurer différemment la répartition de la MEV par blocs de 128 octets entre programme BASIC et formules du tableur. Ainsi, après un EQU#3, l'ordre MEM répond 8958 (soit $9342 - 3 \times 128$), alors que MEM# donne 512 octets.

Pour ce qui est des périphériques, nous ne savons pas si Sharp en commercialisera qui soient spécifiques au 1261. Toujours est-il que notre essai avec le bloc CE-125 s'est avéré concluant. Le branchement ne pose aucun problème, l'imprimante fonctionne toujours, et les micro-cassettes du PC-1251 sont même relues par le 1261.

La machine devrait être commercialisée en France au mois de juillet pour un prix d'environ 2 000 F.

□ CB

■ UNE CASSETTE

Pour le Canon X-07
Fichiers
Logi'Stick
Distribuée par DDI
Prix : environ 130 FF

■ Comme tous les logiciels prêts à l'emploi destinés au X-07, **Fichiers** se présente sous la forme d'une cassette standard. La notice compte une dizaine de pages, en français. La concision est donc de rigueur.

Avec ce logiciel, on peut créer un fichier dans une zone mémoire de travail, puis le « ranger » dans une autre zone afin de rappeler un second fichier. La taille maximale de tout fichier sera environ la moitié de la MEV disponible, ce qui est largement suffisant.

Sont disponibles 13 fonctions (numérotées de 1 à 9 puis appelées C, O, F et R) comportant chacune jusqu'à quatre options. Mais ne vous laissez pas impressionner par ces précisions : les diverses commandes sont accompagnées de menus simples.

Chaque nouveau fichier porte un nom (sur 8 lettres), et contient au plus dix rubriques nommées en 8 lettres mais composées de vingt caractères... si la MEV le permet. Une fois la fiche-type définie, l'entrée des données s'effectue au moyen d'un masque de saisie, avec possibilité de recopie d'une fiche sur une autre. Les fiches peuvent être triées en fonction de n'importe quelle rubrique — numérique ou alphabétique — par ordre croissant ; ou bien encore sélectionnées manuellement par l'intermédiaire d'une rubrique, dans le cas d'une mise à jour par exemple.

L'écran ne permet pas de visualiser le contenu complet d'une fiche, ce dont l'imprimante se charge heureusement : un nouveau masque est défini pour la présentation condensée des fiches. Toutes les modifications effectuées sur le fichier en cours ne sont validées qu'au moment où l'on sauve le fichier sur cassette ou carte RAM, ce qui permet de travailler sans risquer de détruire le fichier original.

Parmi les fonctions classiques, notons le « directory », autrement dit le sommaire des fiches présentes. Ce logiciel, très rapide et pratique, automatise la gestion de la quantité de données qui, chaque jour, encombre nos agendas et feuilles de notes : classer ne signifie plus éparpiller, car le X-07 se charge de tout.

Un reproche : l'utilisateur même expérimenté ne pourra pas modifier facilement le programme de base, car celui-ci est d'une belle complexité.

□ MA

Un module d'extension de fonctions d'entrées/sorties pour HP-41

■ Le module I/O HP 82183 A est en vente depuis le début de l'année 84 à un prix voisin de 730 FF. Il ne nécessite la présence d'aucun des modules auxquels il apporte un complément de fonctions : ces 59 nouvelles fonctions sont regroupées dans quatre rubriques du CATALOG 2.

On y trouve 7 fonctions étendues pour lecteur de cassettes qui permettent de désigner un fichier par son numéro d'ordre, c'est-à-dire de travailler au « coup par coup ». Les 9 fonctions améliorées du module X-fonctions concernent le traitement des chaînes de caractères. Elles ne font pas double emploi avec celles du précédent module, dont elles diffèrent toutes au moins par un détail.

Les 34 fonctions de contrôle de l'interface HP-IL ont donné leur nom au module : entrées/sorties. Elles permettent de contrôler la présence, le fonctionnement et les transmissions de données vers des périphériques divers tels qu'imprimante ou modem, aussi bien au niveau de caractères ASCII, de programmes complets ou de simples octets.

Ces instructions de base sont indispensables pour piloter efficacement des périphériques sophistiqués et rapides.

Les neuf dernières fonctions de contrôle (« avancées ») sélectionnent différents modes de transmission sur la boucle IL, et sont communes au HP-75 C.

Le module d'entrées/sorties prolonge — et métamorphose — l'utilisation de la HP-41, dont le langage continue donc à s'enrichir d'instructions préprogrammées.

□ MA

■ UN LIVRE

Jeux de calculatrices
Robinson du Plessis
MA éditions
Paris, 1983
Broché, 160 pages
Prix : 66 FF

■ Robinson du Plessis est né en 1920, à Las Vegas (ce n'est pas moi qui l'invente, c'est écrit sur la jaquette du livre). Il a donc 63 ans au moment où il se décide à écrire son ouvrage. En réalité, il nous révèle en page 155 sous le titre « Appel à tous : adressez-nous vos programmes » que ce livre est une création collective réalisée à partir de travaux faits dans les clubs,animés par lui-même. J'ignore si ces auteurs (je cite) « auront l'immense gloire de voir leurs noms associés à ceux de Robinson du Plessis et d'Amardi dans un prochain ouvrage », mais ils semblent avoir été quelque peu oubliés dans le premier. En tout cas, l'ouvrage arrive tard. Les calculatrices dont parle du Plessis (Robinson, vous dis-je) sont très classiques : HP-25, HP-67/97, TI-57...

C'est un peu dommage pour ce recueil par ailleurs bien conçu, bien illustré et d'une très bonne lisibilité. Les programmes proposés, en général en versions TI et HP, ne sont pas tous très originaux (c'est le moins que l'on puisse dire). Ils sont par contre convenablement commentés et les textes ne sont pas dépourvus d'humour (Amardi, on aime, ou on n'aime pas !). Pour la plupart, ils sont plutôt orientés vers la simulation du hasard et les jeux. Un ensemble assez cohérent, des illustrations qui égaient l'ouvrage, mais une sortie bien tardive...

☐ BE

■ POUR LES ZX

Pratique des Sinclair ZX 81 et Timex Sinclair T/S 1000
Henri Lilen
Editions Radio
Paris, 1983
Broché, 158 pages
Prix : 80 FF

■ D'un grand format, agréablement présenté, ce livre est destiné à ceux qui, n'ayant aucune connaissance en informatique, ont acheté par curiosité un ZX 81. Il n'est question dans l'ouvrage que du ZX 81 en version de base (1 Ko de mémoire), et la seule extension envisagée est l'imprimante (l'auteur lui consacre environ 8 pages).

Ceux qui se sont déjà initiés au ZX 81 à l'aide du manuel livré avec l'appareil tireront peu de choses de cet ouvrage. Si, en revanche, vous êtes vraiment débutant, si vous désirez un apprentissage très simple et très progressif, cet ouvrage devrait vous convenir. La lecture est facile ; ce qui se passe sur l'écran est encadré, et des commentaires apparaissent, à la manière des bandes dessinées, dans des bulles. Les points particuliers sont mis en évidence par l'utilisation de caractères gras. Mentionnons ainsi les exercices régulièrement proposés pour permettre au lecteur de vérifier s'il a bien assimilé le contenu de chaque chapitre (les solutions sont données en fin de chapitre).

C'est un bon manuel pour le débutant, mais on ne découvrira pas dans ce livre toutes les finesses du Basic Sinclair : ce n'est pas son propos. Pas un commentaire sur les fonctions trigonométriques, sur les PEEK ou les POKE, etc. On reste au niveau de l'initiation, mais de l'initiation bien faite.

☐ BE

Magazine

Découvrez le ZX 81 et le Timex Sinclair T/S 1000
Douglas Hergent
Traduit de l'anglais par Jean-Pierre Loison
Editions Sybex
Paris, 1983
Broché, 196 pages
Prix : 79 FF

■ Voilà un autre ouvrage qui vise le même public que le livre précédent, c'est-à-dire les débutants un peu désorientés par le manuel livré avec l'appareil, et qui recherchent une initiation plus progressive. Une petite différence cependant, signalée dès l'introduction : l'ouvrage est également écrit pour les acheteurs du Timex Sinclair 1000, qui possède 2 Ko de mémoire d'origine. Plusieurs des programmes proposés ne fonctionnent pas sur le ZX 81 de base ; il faudra donc une extension mémoire si l'on veut les utiliser tous.

La conception de l'ouvrage est très particulière : on y trouve, après une longue description physique du ZX et des différentes fonctions accessibles au clavier, un assez joli programme de salutation (11 lignes), mais le lecteur n'est pas invité à le frapper tout de suite. L'apprentissage est fait à partir de ce programme, en reprenant les lignes une à une et en les commentant (les listes sont obtenues grâce à l'imprimante du ZX). C'est une méthode assez originale qui a le mérite d'amener immédiatement le lecteur à un résultat concret. Le programme est ensuite sauvegardé sur cassette, et l'auteur en profite pour parler des logiciels du commerce, et en particulier de VU-CALC : une digression comme un autre ! On attaque ensuite les deux systèmes graphiques du Basic Sinclair, avec de très courts exemples, pour revenir à des instructions plus classiques : IF - THEN et FOR - NEXT. Tout cela est un peu décousu, mais si l'on s'accroche...

Ce n'est qu'à partir du chapitre 4 que l'on aborde les possibilités de calcul du ZX, les chaînes de caractères, etc. On y trouve aussi un long (et séduisant) programme d'histogramme, abondamment commenté dans les pages suivantes. Le chapitre 5, enfin, va un peu plus loin avec la codification des caractères, la vidéo inverse, les nombres aléatoires, tout cela utilisé dans deux courts programmes : devinette et aide à la décision.

Si la formule retenue par l'auteur ne m'a pas semblé très pédagogique, le livre est cependant original, facile à lire, et il offre un tour d'horizon assez complet sur les possibilités du Sinclair.

☐ BE

PC-1500

Le manuel Sharp du langage-machine

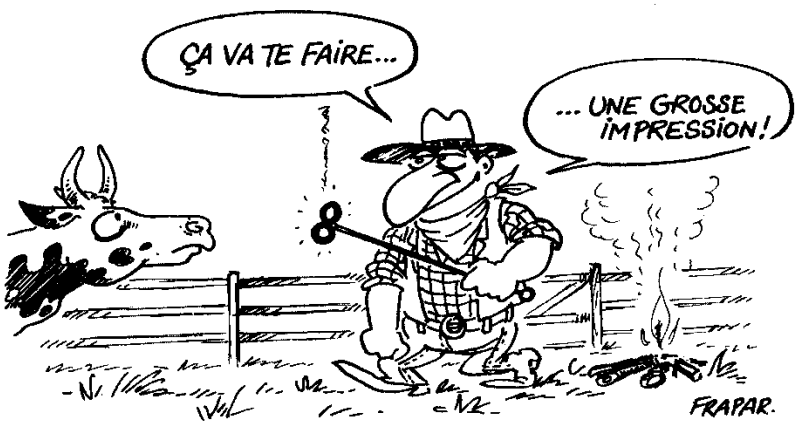
■ Beaucoup de nos lecteurs l'attendaient depuis longtemps. Il est enfin arrivé. Le manuel du langage-machine du PC-1500, édité par **Sharp**, est disponible en version anglaise. A notre connaissance, la seule façon de se le procurer en France est de l'acheter auprès du Club Sharpentier (150 F ttc + 17,40 F de frais de port).

Quant au manuel du langage-machine du 1251, il n'est pas impossible qu'il soit distribué par la même voie dans un avenir proche.

Pour tout renseignement, écrire au club :
SBM (Sharpentier)
151-153 avenue Jean Jaurès
93307 Aubervilliers Cedex

Pour faire grosse impression : les calicots

Le PC-1212 (ou le TRS PC-1) associé à son imprimante est assez pauvre en possibilités graphiques. Les caractères qu'il imprime mesurent moins de trois millimètres de haut... On peut cependant obtenir beaucoup mieux. Voici un moyen de multiplier par treize la taille de ces caractères.



■ Le petit (?) programme dont vous trouverez la liste page ci-contre permet de composer des textes *à la demande*, en caractères géants.

De nombreux essais ayant montré que les caractères d'affiche sont plus lisibles en blanc sur fond sombre plutôt que l'inverse, c'est ce graphisme "négatif" qui a été adopté. Pour le fond, c'est le caractère symétrique le plus couvrant, en l'occurence le "8" qui produit le meilleur contraste. C'est ainsi que les messages obtenus à l'aide du programme proposé offrent une lisibilité optimale à une distance d'un ou plusieurs mètres.

Tout l'alphabet en lettres majuscules

Les caractères sont composés dans une matrice de 12 × 7, inscrite dans un carré de 26 mm de côté. L'intervalle entre deux lettres est de 6 mm, et entre deux mots de 20 mm. La hauteur totale de l'impression est de 30 mm.

Le programme permet de composer les 26 lettres de l'alphabet, en majuscules, plus l'espacement entre les mots. La police de caractères est reproduite ci-contre.

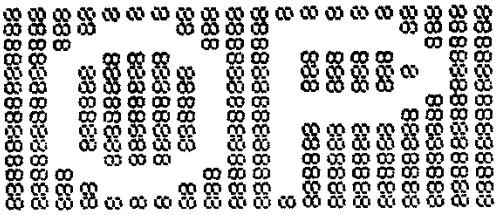
Le mode d'emploi du programme est des plus simples, car votre ordinateur de poche s'est littéralement transformé en machine à écrire :

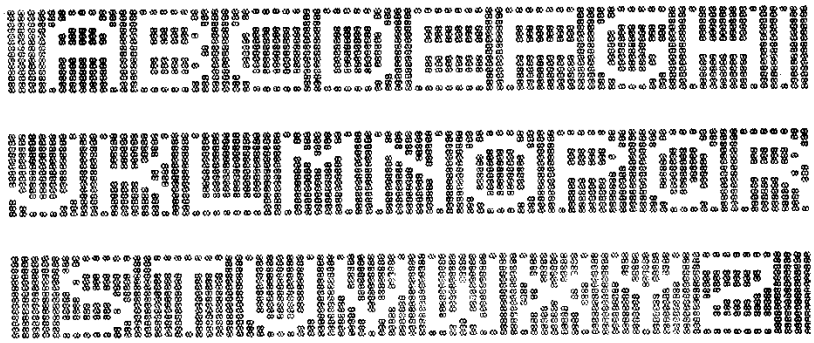
1. Taper RUN ENTER, ce qui a pour effet d'imprimer une bande-amorce et d'initialiser le programme.
2. Après le double bip et l'apparition sur l'afficheur du "?", taper la première lettre du message, puis ENTER. Par exemple, si vous vous appelez PATRICK DUCHARPE et si vous voulez vous offrir une carte de visite de 50 cm de long, tapez P ENTER.
3. Reprendre l'opération n°2 pour chacune des lettres A, T, R, I, C, K.
4. Après le double bip et l'apparition du "?", taper SHFT SPC pour obtenir l'espacement entre deux mots.
5. Reprendre l'opération n°2 pour chacune des lettres D, U, C, H, A, R, P, E.
6. Pour les perfectionnistes : taper un espacement comme pour l'opération n°4 pour obtenir une bande de fin de message.

Maintenant à vous de jouer, mais faites attention aux fautes de frappe !

☐ Pierre Ladislas Gedo

Ci-contre, la 20e et la 21e lettres de l'alphabet grandeur nature.
Ci-dessous, les caractères A à Z après réduction.





Calicots
Programme pour PC-1212 et PC-1
Auteur Pierre Ladislas Gedo
Copyright *l'Ordinateur de poche* et l'auteur

```
 1:GOSUB 10:
   GOSUB 10
 2:" "GOSUB 10:
   GOSUB 10
 3:GOSUB 10:
   GOSUB 10:FOR
   Z=1TO 26:AKZ
   )=Z+40:NEXT
   Z:BEEP 2:
   INPUT Z
 4:A$="8888888"
   :B$="888888
   ":C$="88888
    ":D$="888
     ":E$="8
      "
 5:F$=" 888888"
   :G$="  88888
   ":H$="    88
   8":I$="
    8"
 6:J$="8888  8"
   :K$="888  88
   ":L$="888
   8":M$="88  8
   88":N$="88
     8"
 7:O$="8  8888"
   :P$="8   888
   ":Q$="8    8
   8":R$="8
    8"
 8:S$="88  8  "
   :T$=" 888  8
   ":U$="  8  8
   8"
 9:GOSUB Z:GOTO
   3
10:PRINT "88888
   888888888":
   RETURN
11:PRINT O$;A$:
   RETURN
12:PRINT E$;H$:
   RETURN
13:PRINT E$;I$:
   RETURN
14:PRINT M$;A$:
   RETURN
15:PRINT D$;H$:
   RETURN
16:PRINT D$;I$:
   RETURN
17:PRINT C$;G$:
   RETURN
18:PRINT B$;F$:
   RETURN
19:PRINT A$;G$:
   RETURN
20:PRINT A$;O$:
   RETURN
21:PRINT A$;M$:
   RETURN
22:PRINT A$;K$:
   RETURN
23:PRINT O$;P$:
   RETURN
24:PRINT O$;J$:
   RETURN
25:PRINT Q$;N$:
   RETURN
26:PRINT M$;K$:
   RETURN
27:PRINT B$;" 8
   8  88":
   RETURN
28:PRINT B$;T$:
   RETURN
29:PRINT "8  88
   8 ";T$:
   RETURN
30:GOSUB 28:
   GOSUB 28:
   GOSUB 28:
   RETURN
31:GOSUB 29:
   GOSUB 29:
   GOSUB 29:
   RETURN
32:GOSUB 15:
   GOSUB 26:
   GOSUB 24:
   GOSUB 24:
   RETURN
33:GOSUB 18:
   GOSUB 18:
   GOSUB 18:
   RETURN
34:GOSUB 14:
   GOSUB 11:
   GOSUB 11:
   GOSUB 11:
   GOSUB 14:
   GOSUB 16:
   RETURN
35:GOSUB 13:
   GOSUB 22:
   GOSUB 21:
   GOSUB 20:
   RETURN
36:FOR Y=1TO 3:
   PRINT A$;J$:
   NEXT Y:
   RETURN
41:GOSUB 12:
   GOSUB 27:
   GOSUB 30:
   GOSUB 27:
   GOSUB 12:
   RETURN
42:GOSUB 13:
   GOSUB 29:
   GOSUB 31:
   PRINT G$;U$:
   PRINT L$;F$:
   RETURN
43:GOSUB 32:
   GOSUB 24:
   GOSUB 24:
   GOSUB 24:
   RETURN
44:GOSUB 13:FOR
   Y=1TO 4:
   GOSUB 24:
   NEXT Y:GOSUB
   26:GOSUB 15:
   RETURN
45:GOSUB 13:
   GOSUB 31:
   GOSUB 31:
   RETURN
46:GOSUB 13:
   GOSUB 30:
   GOSUB 30:
   RETURN
47:GOSUB 32:
   GOSUB 24:
   PRINT "88  8
   8 ";T$:PRINT
   D$;T$:RETURN
48:GOSUB 13:
   GOSUB 33:
   GOSUB 18:
   GOSUB 18:
   GOSUB 13:
   RETURN
49:GOSUB 13:
   RETURN
50:PRINT K$;A$:
   GOSUB 34:
   RETURN
51:GOSUB 13:
   GOSUB 33:
   GOSUB 17:
   PRINT J$;O$:
   GOSUB 25:
   RETURN
52:GOSUB 13:FOR
   Y=1TO 6:
   GOSUB 11:
   NEXT Y:
   RETURN
53:GOSUB 35:
   GOSUB 21:
   GOSUB 22:
   GOSUB 13:
   RETURN
54:GOSUB 35:
   GOSUB 19:
   GOSUB 18:
   GOSUB 13:
   RETURN
55:GOSUB 32:
   GOSUB 24:
   GOSUB 26:
   GOSUB 15:
   RETURN
56:GOSUB 13:
   GOSUB 28:
   GOSUB 30:
   PRINT A$;U$:
   PRINT A$;P$:
   RETURN
57:GOSUB 32:
   PRINT O$;J$:
   GOSUB 26:
   GOSUB 12:
   RETURN
58:GOSUB 13:
   GOSUB 30:
   PRINT C$;T$:
   PRINT J$;U$:
   PRINT O$;P$:
   RETURN
59:GOSUB 23:
   PRINT O$;U$:
   GOSUB 31:
   PRINT G$;J$:
   PRINT L$;J$:
   RETURN
60:GOSUB 36:
   GOSUB 13:
   GOSUB 36:
   RETURN
61:GOSUB 16:
   GOSUB 34:
   RETURN
62:PRINT A$;I$:
   PRINT C$;F$:
   PRINT L$;A$:
   PRINT P$;A$:
   PRINT L$;A$:
   PRINT C$;F$:
   PRINT A$;I$:
   RETURN
63:GOSUB 13:
   GOSUB 11:
   GOSUB 14:
   PRINT D$;G$:
   GOSUB 14:
   GOSUB 11:
   GOSUB 13:
   RETURN
64:GOSUB 25:
   PRINT J$;O$:
   GOSUB 17:
   GOSUB 18:
   GOSUB 17:
   PRINT J$;O$:
   GOSUB 25:
   RETURN
65:PRINT A$;N$:
   GOSUB 20:
   GOSUB 19:
   PRINT E$;F$:
   GOSUB 19:
   GOSUB 20:
   PRINT A$;N$:
   RETURN
66:PRINT R$;J$:
   PRINT "8  88
     ";J$:GOSUB
   31:PRINT O$;
   "  88  8":
   PRINT O$;R$:
   RETURN
```

Analyse du programme

Lignes	Commentaires
1-3	Impression de la bande-amorce. Assignation des mémoires numériques : à chaque lettre de l'alphabet est affecté un numéro d'ordre Z égal à son rang + 40. Par exemple, A = 41, B = 42, etc. Introduction de Z.
4-8	Assignation des mémoires alphanumériques : chaque mémoire de A$ à U$ correspond à un élément graphique de 7 éléments.
9	Le pointeur est dirigé vers la ligne portant le numéro d'ordre Z de la lettre choisie, la dessine, introduit un petit intervalle et réassigne les variables numériques de A à Z.
10-29	Sous-programme d'impression de colonnes de 14 éléments.
30-36	Sous-programme d'impression de groupements répétitifs de colonnes.
41-66	Sous-programme d'impression de chaque lettre.

Pour n'oublier aucun espace, il faut savoir que, dans ce programme, chaque variable alphanumérique contient exactement 7 caractères. Le nombre d'espaces se déduit ainsi du nombre de "8" présents.

Par exemple, T$ = " 888 8" soit quatre "8" et, donc, 3 espaces. Le reste se fait à l'œil.

Si on avait beaucoup d'argent...

Parmi les applications les plus fascinantes de l'informatique figurent les simulations en tout genre. Simulation de vol par exemple : vous pilotez un avion mais sans courir de risques. Avec ce programme pour FX-702, vous êtes aux commandes de l'économie d'un pays (imaginaire, rassurez-vous).

■ Modèle, microéconomie, productivité... que de mots barbares. Pour faire leur connaissance, sans souffrir, il existe une solution : le jeu de simulation.

Imaginez que vous gouvernez un pays imaginaire qui ne peut prospérer que grâce à sa population et aux produits de l'agriculture, de l'industrie ou du commerce.

En dix cycles de production, de l'année 0 à l'année 9, vous cherchez à atteindre un profit maximum, évalué en nombre de lingots. Il vous faudra bien sûr tenir compte de critères économiques comme la population, son évolution, l'emploi, le logement, l'équipement en écoles, universités ou hôpitaux et surveiller aussi l'évolution de la production qui découle des investissements, du travail, de la consommation...

Prendre un bon départ

Pour chaque année, le plan général du jeu est le même : affichage des données (tirées au hasard au départ), répartition de vos lingots aux différents secteurs de l'économie (nourriture pour la population, agriculture, industrie, commerce et investissements), enfin répartition de la main-d'œuvre entre l'agriculture, l'industrie et le commerce. Le programme affiche alors les données de l'année suivante qui sont calculées en fonction des résultats de l'année écoulée.

Au départ de chaque cycle, les données affichées sont l'année, le nombre de points (c'est un jeu !), le taux de natalité (c'est un pourcentage, 4 signifiant par exemple que la population augmente de 4 % par an), le montant de vos biens en lingots (PRD pour produit) et le nombre d'habitants de votre pays (POP pour population).

Le programme s'arrête alors sur un point d'interrogation. Il attend l'entrée d'une option : 1, 2 ou 3. L'option 1 vous permet de revoir les données de l'année, elles défilent assez rapidement. L'option 2 vous permet de connaître le niveau de vos biens matériels : A (pour agriculture) est le nombre de terres en exploitation, I (pour industrie) le nombre d'usines, C (pour commerce) le nombre de magasins, L le nombre de logements, E le nombre d'écoles, H le nombre d'hôpitaux et U le nombre d'universités.

Manger pour vivre

L'option 3 permet de continuer le jeu. Une question apparaît alors : NOUR ? Il faut en effet commencer par nourrir la population : chaque personne consomme un lingot par an. Il faut donc compter un lingot par habitant, sans oublier ceux qui vont naître durant l'année.

On doit ensuite acheter des matières premières qui vont servir à la consommation en répondant à

Liste des secteurs

1 A Agriculture
2 I Industrie
3 C Commerce
4 L Logements
5 E Ecoles
6 H Hôpitaux
7 U Universités

Tableau de production	AGRI	IND	COM
Coût de l'unité (en lingots)	10	100	50
Unités de consommation par cycle	1	20	15
Unités de main-d'œuvre par cycle	5	60	10
Unités de production par cycle	10	90	35
Production par unité de main-d'œuvre	2	1,5	3,5
Production par unité de consommation	10	4,5	2,4

L'achat d'une école augmente la productivité dans les trois secteurs de l'économie et l'achat d'une université n'augmente que la productivité de l'industrie.

AGRI ?, IND ?, COM ? Le tableau de production, reproduit précédemment, indique le montant de la consommation par cycle et pour chaque secteur. Enfin, pour que votre pays puisse prospérer, il faut penser aux investissements. A la question INVEST ?, on répond en indiquant le nombre de lingots qui serviront aux divers investissements. Reste alors à répartir ces lingots entre les différents secteurs.

Le nombre que vous avez choisi réapparaît donc suivi de

« AICLEHU ? » où chaque lettre représente un secteur de l'économie acceptant les investissements. Le programme attend alors l'indication du secteur par l'entrée d'un chiffre entre 1 et 7 représentant le rang de la lettre (voir, en encadré, la liste des secteurs).

Pour investir dans le logement (L), par exemple, il faut introduire 4, rang de la lettre L, et répondre à LOG 60 ? Le nombre 60 représentant le coût en lingots d'un logement, la réponse se fait sous la forme d'un multiple de 60. Ainsi, l'achat de trois logements oblige à introduire 180 (et non 3). Tout dépassement du multiple est perdu.

Simulation d'économie
Programme pour FX-702 P
Auteur Pedro Inigo Yanez
Copyright *l'Ordinateur de poche* et l'auteur

Faire DEFM 3 avant de lancer le programme.

```
VAR: 56 PRG: 1440

P0: 132 STEPS
  10 STAT Z:VAC :A4=
     2:B0=1.5:B6=3.5
     :A5=10:B1=4.5:B
     7=2.4:A6=10
  15 B8=35:B2=90:K=4
     :Z=SX:SAC
  20 M=INT (RAN#*250
     )+150:P=INT (RA
     N#*500)+600+M:V
     =0:Q=P
  30 Q=P:C=0:GOTO #1

P1: 508 STEPS
   1 WAIT 20:PRT "AN
     NEE";C,T;" PTS/
     NAT";K,"PRO";P;
     "/POP";M;
   2 INP $:IF $="2";
     WAIT 40:GSB 6:G
     OTO 2
   3 IF $="1" THEN 1
   4 IF $*"3":GSB #5
     :GOTO 2
   5 GOTO 10
   6 PRT "A";A3;"/I"
     ;A9;"/C";B5,"L"
     ;L;"/E";E;"/H";
     H;"/U";U;:RET
  10 GSB #2:T=T+Q
  20 PRT :PRT "M-OE"
     ;M;
  21 INP "AGRI",A1:I
     F A1>A3*5 THEN
     20
  22 INP "IND",A7:IF
      A7>A9*60 THEN
     20
  23 INP "COM",B3:IF
      B3>B5*10 THEN
     20
  24 A=A1+A7+B3:IF A
     >M:GSB #5:GOTO
     20
  35 GSB #4:A=1-A/M:
     IF A≥.3:PRT "GR
     EVES":T=T-(T*.1
     5)
  40 M=M+INT (M*(K/1
     00)):O=L*200:IF
      M>O:PRT M-O;"S
     ORTIS":T=T-M+O:
     M=O
  45 IF M>N:PRT N-N;
     "SORTIS":T=T-M+
     N:M=N
  50 K=K+2:C=C+1:IF
     C<10 THEN 1
  60 WAIT 60:PRT "AN
     NEE";C,"TOTAL:"
     ;T
  70 IF T>Z:Z=T
  80 PRT " SCORE MAX
     :";Z:STOP :GOTO
      60

P2: 103 STEPS
  10 PRT P;:INP " NO
     UR",N,"AGRI",A2
     ,"IND",A8,"COM"
     ,B4,"INVST",I
  11 Q=Q-N-A2-A8-B4-
     I:IF Q<0:Q=P:GS
     B #5:GOTO 10
  12 X=0:GSB #3:RET

P3: 523 STEPS
   1 I=I-X:PRT I;:IF
      I≠0:RET
   2 INP " AICLEHU",
     Z:IF Z≥1;IF Z≤7
     ;GSB 3+(Z*3):GO
     TO 1
   3 GSB #5:GOTO 1
   4 Y=INT (X/A):IF
     I-X≥0:RET
   5 X=0:Y=0:PRT "ER
     R";:RET
   6 A=10:INP "AGR 1
     0",X:GSB 4:A3=A
     3+Y:RET
   9 A=100:INP "IND
     100",X:GSB 4:IF
      A9+Y>INT (A3/5
     ) THEN 5
  10 A9=A9+Y:K=K-Y*2
     :RET
  12 A=50:INP "COM 5
     0",X:GSB 4:IF B
     5+Y>INT (A9/2)
     THEN 5
  13 B5=B5+Y:RET
  15 A=60:INP "LOG 6
     0",X:GSB 4:L=L+
     Y:K=K+5*Y:RET
  18 A=100:INP "ECO
     100",X:GSB 4
  19 E=E+Y:A=Y*.5:A4
     =A4+A:B0=B0+A:B
     6=B6+A:A5=A5+A*
     5:B1=B1+A*3
  20 B7=B7+A:A6=A6+A
     *5:B2=B2+60*A:B
     8=B8+10*A:K=K-3
     :RET
  21 A=200:INP "HOP
     200",X:GSB 4:H=
     H+Y:K=K+10*Y:RE
     T
  24 A=250:INP "UNI
     250",X:GSB 4:U=
     U+Y:B0=B0+Y:B1=
     B1+3*Y
  25 B2=B2+Y*60:RET

P4: 101 STEPS
  10 P=0:FOR X=0 TO
     12 STEP 6
  11 FOR Y=1 TO 3:A(
     Y+18)=A(Y+X)*A(
     Y+X+3):NEXT Y:Q
     =B9
  12 IF C0<Q:Q=C0
  13 IF C1<Q:Q=C1
  14 P=P+INT Q:NEXT
     X:Q=P:RET

P5: 22 STEPS
   1 WAIT 50:PRT "IM
     POSSIBLE":RET
```

Structure du programme

1. Initialisation des variables (P0)
2. Affichage des résultats (P1)
3. Répartition de la production (P2)
4. Répartition des investissements (P3)
5. Répartition de la main-d'œuvre (P1)
6. Calculs de la production (P4)
7. Calculs de la population (P1)
8. Affichage des résultats si 10e année (P1)

Sinon retour en 2

Si on avait beaucoup d'argent...

A propos de

LIST

avez-vous lu notre éditorial en page 5 ?

Bases de calcul

Production

Le calcul de la production est fonction, ici, de trois facteurs : la main-d'œuvre, la consommation (ou matières premières) et les unités de production. Soit P_f la production de chacun de ces facteurs :

P_f = (nombre d'unités du facteur) × (productivité de ce facteur).

La production de chaque secteur (agriculture, industrie, commerce) sera égale à la plus petite des trois valeurs calculées P_f.

La production totale sera égale à la somme des productions des trois secteurs.

Population

Chaque année, à la population initiale s'ajoute une population potentielle p :

p = (population initiale) × (taux de natalité).

La population logée l est :

l = (nombre de logements) × 200.

La population nourrie doit inclure celle qui est née dans l'année (population potentielle).

La population finale est celle qui est à la fois nourrie et logée.

A ce niveau, on doit avoir bien présent à l'esprit quelques points importants :

- l'achat de logements est obligatoire, un logement abrite 200 personnes et chaque personne non logée quitte le jeu au cycle suivant (cela vaut aussi pour les personnes nées pendant le cycle en cours) ;
- l'achat de certains biens influe sur le taux de natalité. Les usines, par exemple, tendent à faire diminuer ce taux et les hôpitaux à le faire augmenter ;
- l'achat d'écoles ou d'universités accroît la productivité ;
- l'agriculture, l'industrie et le commerce enfin sont les seuls secteurs de l'économie qui augmentent directement le produit. Ils sont soumis à deux contraintes particulières : une usine ne peut être acquise qu'après cinq terres et le commerce, lui, ne peut être acquis qu'après deux usines. Les autres investissements sont libres.

Pour faire progresser votre produit, vous n'êtes pas obligé de pourvoir tous les postes. Quant à la main-d'œuvre, elle doit être répartie entre l'agriculture, l'industrie et le commerce. Attention au chômage, il influe directement sur la production : un taux de chômage supérieur à 30 % fait perdre 15 % des profits... Il faut donc être très attentif à l'emploi.

Il se peut d'ailleurs que le programme annonce des « grèves » dues à un chômage trop important. Ces grèves entraînent un certain nombre de « sortis » : ce sont les habitants qui quittent le jeu parce qu'ils ne sont pas nourris ou pas logés. Leur départ fait directement diminuer votre profit.

Chacun son lingot

Si vous atteignez l'année 10, c'est-à-dire si vous avez pu garder PRD et POP strictement positifs pendant 10 cycles, alors le total de vos points apparaîtra suivi du score maximum atteint à ce jour.

Le but du jeu n'est pas de créer beaucoup d'emplois ou de posséder beaucoup de moyens de production. Il est de maximiser le profit. La faillite vous guette dès que votre produit (PRD) est nul.

Les salaires sont invariables et égaux pour tous : ils s'élèvent à un lingot par personne et par an. Pour recommencer une partie, on appelle P0, et pour reprendre une partie interrompue, P1. Des calculs peuvent être faits en cours de jeu, à la suite de chaque instruction INP. Un appui sur EXE allume l'indicateur STOP ; on effectue normalement les calculs au clavier, puis on reprend le programme en appuyant sur CONT.

Ainsi, vous aurez « dirigé » un marché vaste mais imaginaire. En fait, cette simulation ne rend pas compte de la réalité qui est beaucoup plus compliquée que cela.

☐ Pedro Inigo Yanez

Pour être sûr d'en sortir...

Prendre le risque d'entrer dans un labyrinthe, c'est faire preuve d'un peu de témérité ou de beaucoup d'inconscience. A moins que vous ne connaissiez tous les recoins du dédale, mais c'est un privilège dont seul jouit l'architecte.
(Programme pour PC-1251.)

■ Voulez-vous devenir architecte ? Construire les plans d'un dédale où vos amis iraient s'aventurer, c'est possible, mais à condition de passer vous aussi par cette épreuve et d'en sortir victorieux.

Munissez-vous d'une feuille de papier quadrillée sur laquelle vous tracerez les contours d'une grille 10 × 10 dont les colonnes et les lignes seront numérotées de 0 à 9. Introduisez le programme dans votre PC-1251 (voir page suivante). Vous êtes prêt ? Allons-y : tapez DEF = pour l'initialisation. Vous voilà projeté au cœur du labyrinthe, en case 44. Le chiffre des dizaines est celui de la ligne, le chiffre des unités celui de la colonne.

Des couloirs sans issue...

Cette première salle vous ouvre tous les horizons. Les lignes blanches représentent les couloirs que vous pouvez emprunter et toutes les directions sont possibles. Pour vous déplacer, vous utiliserez le pavé numérique : 8 pour aller au Nord, 6 pour l'Est, 4 pour l'Ouest et 2 pour le Sud.

Après quelques déplacements prudents, vous vous apercevrez vite que le parcours vous réserve des surprises : pas de rencontres désagréables, mais des portes qui se referment derrière vous, des couloirs sans issue et... quelques oubliettes.

Retrouver les issues de secours

Ne croyez pas que les bords de la grille limitent votre espace de jeu. Arrivé en case 89, essayez par exemple de continuer vers l'Est et vous serez aussitôt transporté par voie souterraine en case 90. De même, aller vers l'Ouest en 60 équivaut à faire un petit voyage en métro jusqu'en 59. En revanche, dépasser les limites qui vous sont fixées au Nord et au Sud vous ramènera invariablement au point de départ en case 44.

Bonne chance, la sortie existe et pour la trouver, n'hésitez pas à vous aider de votre grille et d'un crayon.

A votre tour d'égarer les autres

En tapant le programme, vous avez certainement remarqué que les premières lignes débutaient par des DATA. Ces DATA sont suivis des codes attribués à chacune des cases du labyrinthe. Il suffira de les rem-

Combinaison	Code
↑	1
→|	2
↓	3
←|	4
↑→	5
↑↓	6
↑←	7
→↓	8
←→	9
←↓	10
←↑→	11
↑→↓	12
←→↓	13
←↓↑	14
←↑→↓	15

Quinze combinaisons de sortie

N
O ← → E
S

Pour être sûr d'en sortir...

placer par vos propres codes, et de quelques autres modifications, pour redéfinir complètement l'implantation des salles et des couloirs.

Il existe pour chaque case quinze combinaisons possibles : la pièce peut avoir par exemple une seule porte au Nord, ou bien deux, une à l'Est et l'autre au Sud, elle peut encore s'ouvrir à tous les vents, sur ses quatre côtés. A vous de décider. Consultez le tableau des combinaisons de sortie et choisissez.

Le programme mettra en B$(0) le code de votre première pièce, en B$(1) celui de la deuxième et ainsi de suite. Il faut aussi changer quelques paramètres en lignes 10, 15, 20, 50, 54 et 116, comme indiqué dans l'encadré ci-contre.

Si le nombre de colonnes et de lignes est supérieur ou inférieur à dix, le numéro des cases sera celui de leur place sur la grille. Dans le cas d'un parcours de neuf pièces, celles-ci seront distribuées comme suit :

0	1	2
3	4	5
6	7	8

Une précision intéressante : si votre labyrinthe occupe une grille 10 × 10, il vous reste environ 2 200 octets. Si votre imagination ne vous fait pas défaut, cela vous permettra de rajouter des niveaux, des monstres, des pièges et autres embuscades de toutes sortes et vous obtiendrez alors un véritable jeu d'aventure.

☐ Etienne Maetz

Les lignes à modifier

Ligne 10 :
" = " CLEAR : DIM B$ (**99**) * 2 : FOR I = 0 TO **99** : READ B : B$(I) = STR$ B : NEXT I : A = **44** : U = 62...

Dans cette ligne, 99 correspond au nombre total de salles moins une, dans le cas d'une grille 10 × 10 (100 − 1). Pour une grille 3 × 3, ce sera 8 (9 − 1).

Dans la variable A se trouve le numéro de la case départ.

Ligne 15 :
IF A<0 OR A>**99** LET A = **44**

où 99 représente le nombre total de cases moins une, et la variable A le numéro de la case où l'on se retrouve lors du dépassement des limites inférieures et supérieures de la grille.

Ligne 20 :
IF A = **0** PAUSE « BRAVO !!! »...

Variable A : numéro de la case de sortie, OUF !

Lignes 50 et 54 :
IF B = 8 LET A = A − **10**
IF B = 2 LET A = A + **10**

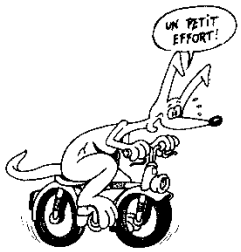
où 10 est le nombre de colonnes du labyrinthe

Ligne 116 :
"A" A = **44** : GOTO 25

Variable A : numéro de la case où l'on reprend une nouvelle partie.
Pour rejouer, faire DEF A.

Labyrinthe
Programme pour PC-1251
Auteur Etienne Maetz
Copyright l'*Ordinateur de poche* et l'auteur

```
  2:DATA 6,12,3,8,14,5,5
    ,13,5,4,12,12,11,14,
    5,13,6,14,10,9,11,2,
    8,1,3,7,9,6,13,9,12,
    12
  3:DATA 3,6,15,10,7,7,9
    ,5,3,8,1,14,15,13,9,
    5,3,15,6,15,11,2,6,3
    ,2,11,4,15,11,8,3,8
  4:DATA 10,9,6,12,6,13,
    7,4,3,4,10,4,13,14,3
    ,3,15,14,12,8,11,10,
    12,8,12,12,15,6,2,8
  5:DATA 13,10,5,12,6,12
 10:"=" CLEAR : DIM B$(9
    9)*2: FOR I=0 TO 99:
    READ B:B$(I)= STR$ B
    : NEXT I:A=44:U=62:T
    =54:V=48:F=50:G=30:
    BEEP 1
 11:RESTORE
 15:IF A<0 OR A>99 LET A
    =44
 20:IF A=0 PAUSE "BRAVO
    !!!": PAUSE " VOUS E
    TES SORTI !!!": END
 25:GOSUB 100+ VAL B$(A)
 30:WAIT 0: PRINT "
    "; STR$ A: CALL &11E
    0
 35:POKE &F828,X,X,Y,Z,Z
 40:B= VAL INKEY$ : IF B
    =0 THEN 40
 45:GOTO 60+2*( VAL B$(A
    )-1)
 50:IF B=8 LET A=A-10
 52:IF B=6 LET A=A+1
 54:IF B=2 LET A=A+10
 56:IF B=4 LET A=A-1
 58:GOTO 15
 60:IF B=8 GOTO F
 61:GOTO G
 62:IF B=6 GOTO F
 63:GOTO G
 64:IF B=2 GOTO F
 65:GOTO G
 66:IF B=4 GOTO F
 67:GOTO G
 68:IF B=8 OR B=6 GOTO F
 69:GOTO G
 70:IF B=8 OR B=2 GOTO F
 71:GOTO G
 72:IF B=8 OR B=4 GOTO F
 73:GOTO G
 74:IF B=6 OR B=2 GOTO F
 75:GOTO G
 76:IF B=6 OR B=4 GOTO F
 77:GOTO G
 78:IF B=2 OR B=4 GOTO F
 79:GOTO G
 80:IF B<>2 GOTO F
 81:GOTO G
 82:IF B<>4 GOTO F
 83:GOTO G
 84:IF B<>8 GOTO F
 85:GOTO G
 86:IF B<>6 GOTO F
 87:GOTO G
 88:GOTO F
101:X=U:Y=V:Z=U: RETURN
102:X=U:Y=T:Z=T: RETURN
103:X=U:Y=6:Z=U: RETURN
104:X=T:Y=T:Z=U: RETURN
105:X=U:Y=V:Z=T: RETURN
106:X=U:Y=0:Z=U: RETURN
107:X=T:Y=V:Z=U: RETURN
108:X=U:Y=6:Z=T: RETURN
109:X=T:Y=T:Z=T: RETURN
110:X=T:Y=6:Z=U: RETURN
111:X=T:Y=V:Z=T: RETURN
112:X=U:Y=0:Z=T: RETURN
113:X=T:Y=6:Z=T: RETURN
114:X=T:Y=0:Z=U: RETURN
115:X=T:Y=0:Z=T: RETURN
116:"A"A=44: GOTO 25
```



Adressage indirect et Basic : la souplesse

L'adressage indirect n'est pas la partie la plus évidente de la programmation. Certains langages s'y prêtent mieux que d'autres. Avec les versions de poche du Basic, c'est un outil à la fois puissant et relativement facile à assimiler.

■ Le Basic de poche est un langage de programmation dont on apprend les rudiments assez vite. En effet, il met en œuvre une syntaxe plus proche du raisonnement habituel que le Langage Machine Spécialisé (LMS que l'on trouve, par exemple, sur les calculatrices TI et HP).

Cette différence est particulièrement nette concernant l'adressage indirect. On sait que le LMS (notation algébrique ou polonaise inverse) conduit parfois l'utilisateur à effectuer des acrobaties pour parvenir à ses fins. Ces acrobaties, même si elles font la fierté du programmeur, sont difficiles à saisir pour le débutant.

Voyons quelles commodités offre de ce point de vue le Basic des poquettes et commençons par un bref rappel sur le principe de l'adressage indirect : on utilise une variable non pas pour l'indication qu'elle contient, mais pour savoir où se trouve l'information recherchée.

En Basic, ce type d'adressage peut le plus souvent s'appliquer à toute instruction nécessitant un argument, utilisation des variables, branchements, boucles, fonctions mathématiques, instructions d'entrée/sortie... et la liste n'est pas complète.

Un des usages les plus courants de l'adressage indirect en Basic est la manipulation des tableaux, numériques ou non. Ceux-ci pourront comporter une, deux dimensions, et parfois plus. Sur certains poquettes, on devra dimensionner les tableaux au préalable (instruction DIM) ou étendre la zone des données. Avec DIM T(5), on établit un tableau comportant six éléments : T(0), T(1), T(2), T(3), T(4) et T(5). Autre exemple, DIM B$ (2,3) ménage un tableau à deux dimensions dont les variables sont :

B$ (0,0) B$ (0,1) B$ (0,2) B$ (0,3)
B$ (1,0) B$ (1,1) B$ (1,2) B$ (1,3)
B$ (2,0) B$ (2,1) B$ (2,2) B$ (2,3)

On remarque que sur la plupart des ordinateurs, l'instruction DIM T(n) réserve en mémoire n+1 emplacements. Les T(i) sont aussi appelés variables indicées (pour i variant de 0 à n).

En voici une illustration simple :

```
 5  S=0 : DIM T(9)
10  FOR I = 0 TO 9
15  INPUT "N=?" ; T(I)
20  S = S + T(I)
25  NEXT I
30  PRINT S
35  END
```

Ce programme initialise un tableau unidimensionnel de dix éléments et demande (boucle des lignes 10 à 25) l'introduction de dix valeurs qui sont successivement stockées en T(0), T(1), T(2),..., et T(9). La ligne 20 cumule en S la somme des termes au fur et à mesure de leur introduction, laquelle somme est affichée en ligne 30.

Les personnes qui programment en LMS apprécieront la forme T(I), bien plus parlante que les STO Ind ou RCL Ind...

D'autre part, à la ligne 15 (et cela vaut pour la plupart des poquettes Basic), on peut non seulement entrer un nombre, mais aussi une expression combinant variables, nombres et fonctions mathématiques : 5 × 2, 5 × A ou 8 + √2/(X+Y). L'expression sera calculée, à condition qu'elle soit conforme à la syntaxe du Basic, et sa valeur sera rangée en T(I). Cette propriété, très pratique, est pour ainsi dire un privilège des poquettes ; on la retrouve sur peu d'ordinateurs de table.

Une chance sur six...

Les expressions calculées peuvent aussi servir d'arguments pour l'adressage indirect, ce qui accroît encore la puissance de ce dernier. Rien n'empêche, en effet, d'utiliser des formules aussi compliquées que : PRINT A (INT ABS (COS(I) * 100)).

Voici, avec une version de la roulette russe — parfaitement inoffensive —, un exemple plus simple.

Sur Sharp PC-1251 :

```
 5  DIM P$ (6)
10  P$ (RND 6) = "PAN !"
15  FOR I = 1 TO 6
20  PRINT P$ (I)
25  NEXT I
```

Version Casio FX-702 P (faire DEFM 1) :

```
 5  VAC
10  A$ (RAN#*6+1) = "PAN !"
15  FOR I = 1 TO 6
20  PRT A$ (I)
25  NEXT I
```

L'ordinateur remplace ici un revolver "six coups" dont le barillet est simulé par un tableau à une dimension, P$ (n) ou A$ (n). La ligne 10 stocke la chaîne alphanumérique "PAN !" dans une variable du

Adressage indirect et Basic : la souplesse

LIST
vous donne rendez-vous
fin juin
chez votre marchand de journaux

tableau dont l'indice est obtenu par le générateur de nombres aléatoires de l'ordinateur, soit ici un indice de 1 à 6 compris.

Une fois la balle placée dans un des logements du barillet, il ne reste plus aux joueurs qu'à se passer la machine à tour de rôle. La boucle des lignes 15 à 25 finira tôt ou tard par faire apparaître le "PAN !". Pour presser sur la détente, on appuie sur ENTER ou sur CONT. Celui qui lance le programme tire le premier.

D'une ligne à l'autre

Le Basic permet d'effectuer des branchements du type GOTO ou GOSUB, mais avec nettement plus de souplesse que les LMS, même sophistiqués. Pour nous en convaincre, examinons l'exemple suivant :

Version Sharp PC-1251

```
 5  INPUT "N=?" ; N
10  IF N>3 OR N<1 THEN END
15  GOSUB 100+N : PRINT
    "FIN" : END
101 PAUSE "LIGNE 101" : RETURN
102 PAUSE "LIGNE 102" : RETURN
103 PAUSE "LIGNE 103" : RETURN
```

Version Casio FX-702 P

```
 5  INP "N=", N : WAIT 30
10  IF N>3 ; END
11  IF N<1 ; END
15  GSB 100+N : PRT "FIN" :
    END
101 PRT "LIGNE 101" : RET
102 PRT "LIGNE 102" : RET
103 PRT "LIGNE 103" : RET
```

Le programme demande à l'utilisateur d'entrer un nombre compris entre 1 et 3. La ligne 10 (les lignes 10 et 11 pour le 702 P) vérifie cette condition et, en cas de succès, l'instruction GOSUB 100 + N est exécutée. La ligne à laquelle se fait le branchement est donc 101, 102 ou 103 selon la valeur de N. Le programme affiche alors le numéro de cette ligne puis retourne en 15 pour afficher "FIN". En utilisant GOTO, le programme précédent aurait pu s'écrire :

Version Sharp PC-1251

```
 5  INPUT "N=?" ; N : A=20
10  IF N>3 OR N<1 THEN END
15  GOTO 100 + N
20  PRINT "FIN" : END
101 PAUSE "LIGNE 101" : GOTO A
102 PAUSE "LIGNE 102" : GOTO A
103 PAUSE "LIGNE 103" : GOTO A
```

Version Casio FX-702 P

```
 5  INP "N=", N : A=20 : WAIT 30
10  IF N>3 ; END
11  IF N<1 ; END
15  GOTO 100 + N
20  PRT "FIN" : END
101 PRT "LIGNE 101" : GOTO A
102 PRT "LIGNE 102" : GOTO A
103 PRT "LIGNE 103" : GOTO A
```

Au programmeur de choisir la méthode la mieux adaptée à ce qu'il recherche : lisibilité, rapidité d'exécution ou faible consommation en octets.

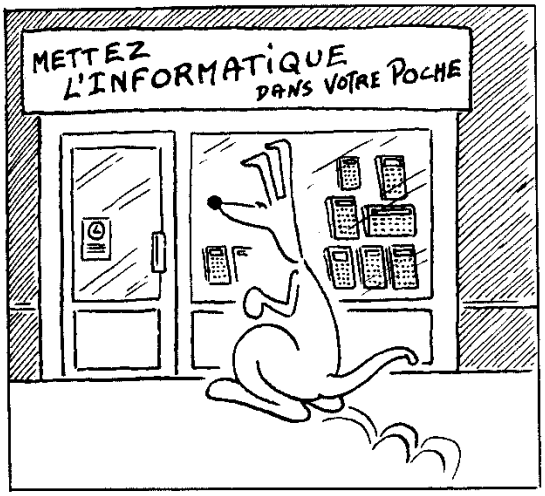
Comme nous l'avons vu plus haut, des expressions comprenant des variables et des fonctions mathématiques peuvent être utilisées en argument de GOTO et de GOSUB. D'autre part, les étiquettes alphabétiques (que l'on trouve sur les poquettes Sharp) se prêtent aussi à l'adressage indirect :

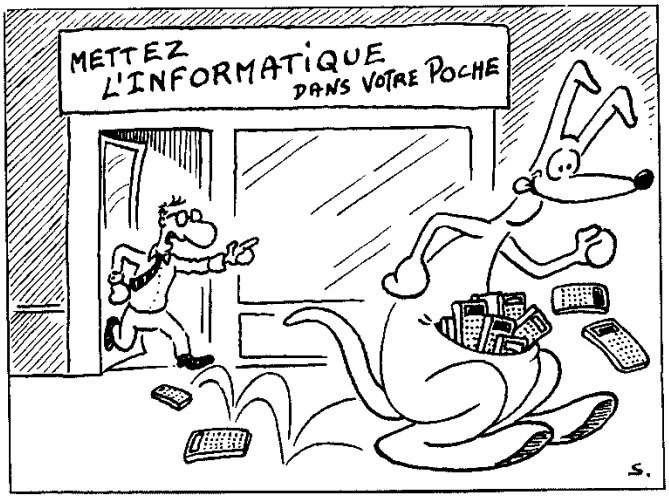
```
  5  INPUT "NOMBRE D'ARTICLES
     ?" ; N
 10  L$ = "N = 1"
 15  IF  N>1  LET  L$="N>1"
(...)
100  GOTO L$
(...)
200  "N = 1" : PRINT "UN SEUL
     ARTICLE" : END
210  "N>1" : PRINT " PLUSIEURS
     ARTICLES" : END
```

Cet exemple n'a d'autre intérêt que de montrer les possibilités inattendues de certaines versions du Basic. D'une façon générale, l'adressage indirect s'applique à toutes les fonctions du poquette travaillant sur des nombres, des chaînes et les variables correspondantes.

Avec un peu d'ordre et de soin, on parvient à des expressions très claires. Mais rien n'empêche évidemment d'utiliser ce même adressage indirect pour rendre un programme illisible !

☐ Jean-Charles Lemasson





Le test de Fermat
(version décomposée)

« Toujours plus loin, toujours plus vite » devrait être la devise de tous ceux qui s'adonnent à la chasse pacifique aux nombres premiers. C'est en application de cette maxime que nous allons revenir sur le test de Fermat (1) grâce à un algorithme particulièrement rapide (exemples sur PC-1211 et PB-100)

■ Rappelons que d'après le « petit théorème de Fermat », si N est un nombre premier et B un nombre quelconque non divisible par N, alors $B^{N-1} - 1$ est divisible par N. La deuxième partie de cette proposition peut également s'écrire $B^{N-1} - 1 \equiv 0 \pmod N$ ou $B^{N-1} \equiv 1 \pmod N$ (prononcer : B puissance N moins un est congru à un modulo N).

L'important c'est le reste

Ces relations signifient simplement que le reste de la division de B^{N-1} par N est égal à 1. Par exemple, pour B = 2 et N = 11 : $B^{N-1} = 2^{10} = 1\,024$ et on peut vérifier que $(B^{N-1} - 1)$ qui vaut 1 023 est effectivement divisible par 11, ou (conclusion équivalente) que le reste de la division de 1 024 par 11 est bien égal à 1. En effet, 1 024 = 11 × 93 + 1.

Cet exemple nous conduit à admettre que le reste de la division de B^{N-1} par N (ce reste ou résidu du test de Fermat sera appelé R, ici, B étant la base) peut fournir des indications dans la recherche des nombres premiers. Attention cependant, car le test de Fermat permet d'affirmer que si N est premier, R est égal à 1, mais il serait hélas faux d'en conclure que si R est égal à 1, N est premier. Il en est *presque toujours* ainsi, mais pas *toujours*.

Voulez-vous un exemple, ou plutôt un contre-exemple ? Prenons B = 4 et N = 15 et calculons le reste R de 4^{14} divisé par 15 :
268 435 456/15 = 17 895 697 ; reste 1.

Dans cet exemple, R est bien égal à 1 et 15 n'est pas premier, bien qu'il « passe le test de Fermat ». C'est pour cette raison qu'il est dit pseudo-premier.

L'existence de ces nombres particuliers, comme 15 dans notre exemple, n'a été découverte qu'au début du dix-neuvième siècle. Cela dit, les Chinois utilisaient, il y a plus de deux millénaires, un test de primarité équivalent à celui de Fermat, et cela dans les deux sens en quelque sorte, sans restriction quant à la réciproque, et sans se rendre compte qu'il fonctionnait dans un sens et pas dans l'autre.

Plus près de nous, les programmes que nous vous présentons reposent sur une observation élémentaire que nous allons illustrer par un exemple. Le plus petit nombre pseudo-premier en base 2 (ici, base est pris au sens de « base du test ») est 341. Tout ce que nous savons de ce nombre est que 2^{340} divisé par 341 donne un reste R égal à 1.

Au lieu d'étudier la valeur de l'indicateur R, examinons celle de R' = R − 1, autrement dit cherchons le reste de la division de $(2^{340} - 1)$ par 341.

Si R' est égal à zéro, N est premier ou pseudo-premier. Mais on

Différence de deux carrés

Certains calculs aux apparences compliquées peuvent être facilités par l'utilisation de formules algébriques : les identités remarquables.

Ce sont des égalités plus ou moins faciles à retenir, souvent très pratiques et qui peuvent être démontrées assez rapidement.

L'une d'entre elles est appliquée ici :
$a^2 - b^2 = (a - b)(a + b)$ où a et b sont des entiers.

En effet, si on remplace a par 2^{170} et b par 1, cette identité devient : $(2^{170})^2 - 1^2 = (2^{170} - 1)(2^{170} + 1)$

Et justement $(2^{170})^2$ est égal à $2^{170 \times 2}$, soit à 2^{340}. Quant à 1^2, c'est bien égal à 1.

(1) *Voir* l'Op *n° 22 page 21.*

Nombres premiers le test de Fermat

peut décomposer $2^{340} - 1$ en facteurs (voir l'encadré « différence de deux carrés »). En effet : $2^{340} - 1 = (2^{170} - 1)(2^{170} + 1)$ et de même : $2^{170} - 1 = (2^{85} - 1)(2^{85} + 1)$ et finalement : $2^{340} - 1 = (2^{85} - 1)(2^{85} + 1)(2^{170} + 1)$.

Pour que le reste de la division de $2^{340} - 1$ par 341 soit égal à zéro, il suffit que l'un des trois facteurs entre parenthèses soit congru à zéro modulo 341. Or $2^{85} \equiv 32 \pmod{341}$ et $2^{170} \equiv 1 \pmod{341}$.

Vous pouvez vérifier ces résultats à l'aide du petit utilitaire ci-dessous, qui retourne, pour chaque nombre Z en base B modulo N, la valeur du reste de la division de B^Z par N. On lance le programme par SHFT = sur le PC-1211 (et par P0 sur le PB-100, si le programme est en P0).

On a donc : $2^{85} - 1 \equiv 31 \pmod{341}$, $2^{85} + 1 \equiv 33 \pmod{341}$ et $2^{170} + 1 \equiv 2 \pmod{341}$.

Facteurs de restes

Aucun de ces facteurs n'est congru à zéro, modulo 341. Autrement dit, quand on divise chacun de ces facteurs par 341, le reste n'est jamais nul. En revanche, le produit des trois restes ($31 \times 33 \times 2 = 2\,046$) est bien un multiple de 341 ; il est donc congru à zéro modulo 341.

Il résulte de cet exemple que l'on

Petit utilitaire : reste R où $B^Z \equiv R \pmod{N}$
Programme pour PC-1211
Auteur Pierre Ladislas Gedo
Copyright *l'Ordinateur de poche* et l'auteur

```
700:"="INPUT "Z=
    ";Z,"N=";N,"
    B=";B
710:E=INT (LOG (
    Z+.5)/LOG 2)
    :C=2^E:D=Z-C
    :R=B
720:FOR W=1TO E
730:R=R*R-N*INT
    (R*R/N):C=C/
    2
740:IF C<=DLET R
    =B*R-N*INT (
    B*R/N):D=D-C
750:NEXT W
760:PRINT "R=";R
```

Pour adapter cet utilitaire au PB-100, tenir compte des remarques accompagnant les programmes 1 et 2.

Test de Fermat décomposé
Programme 1, pour PC-1211
Auteur Pierre Ladislas Gedo
Copyright *l'Ordinateur de poche* et l'auteur

```
10:"A"INPUT " N
   = ";N
20:IF (N>1)*(N<
   E10)*(N=INT
   N)=0BEEP 4:
   GOTO 10
30:IF N=2GOTO 1
   40
40:IF N/2=INT (
   N/2)GOTO 130
50:M=N-1:E=INT
   (LOG (M+.5)/
   LOG 2):C=2^E
   :D=M-C:R=2
60:IF N>=E6LET
   F=E-5:G=INT
   (F*N):H=F*N-
   G
70:GOSUB 200
80:IF C<=DLET R
   =2*R-N*INT (
   2*R/N):D=D-C
90:IF D>0GOTO 7
   0
100:IF R=1GOTO 1
   40
110:IF R=MGOTO 1
   40
120:IF C>2GOSUB
   200:GOTO 110
130:BEEP 2:PRINT
   N;"COMPOSE":
   END
140:J$=" (!?)"
150:IF N<2047LET
   J$=" "
160:BEEP 2:PRINT
   N;"PREMIER";
   J$:END
200:IF R<E6LET R
   =RR-N*INT (R
   R/N):C=C/2:
   RETURN
210:S=INT (FR):T
   =FR-S:U=INT
   (RR/N):V=INT
   (FU):W=FU-V
220:R=E10*(SS-GV
   +2ST-GW-HV+T
   T-HW)
230:IF R<0LET R=
   R+N
240:IF R>=NLET R
   =E10*(SS-GV+
   2ST-GW-HV+TT
   -HW)-N
250:C=C/2:RETURN
```

Pour adapter ce programme au PB-100, il suffit de respecter le Basic de cette machine.

Ainsi, l'étiquette "A" (de la ligne 10) doit être supprimée, l'instruction INPUT doit être suivie d'une virgule (au lieu d'un point-virgule, sur le PC-1211), un IF ne peut être suivi que par THEN ou par un point-virgule, E 6 (ou E 10, E^{-5}, etc.) doit être écrit 1 E 6 (ou 1 E 10, 1 E^{-5}, etc.).

Les instructions BEEP doivent être supprimées (le bip n'existe pas sur PB-100) et la multiplication doit toujours être explicitée (par exemple, RR compris par le PC-1211 s'écrit R * R sur le PB-100). Quant à la ligne 20, il faut la remplacer par les deux lignes : 20 IF N > 1 ; IF N < 1 E 10 ; IF N = INT N THEN 30 et 25 GOTO 10. Enfin, le signe "^" de la ligne 50 est celui de l'élévation à une puissance ("*" sur le PB-100).

peut définir un nouveau test de primarité plus restrictif que le test classique, en effectuant un test sur chacun des facteurs de la décomposition. Notez que ce nouveau type de test, que nous appellerons décomposé, n'admet pas *un* indicateur, mais *plusieurs* indicateurs : le reste de chaque facteur est examiné séparément et si aucun reste n'est égal à zéro, N est composé ; en revanche, dès qu'un reste est égal à zéro, il est inutile de poursuivre les calculs, car N est alors premier ou pseudo-premier.

Ce test décomposé n'est applicable qu'aux nombres impairs, car si N est pair, B^{N-1} ne peut pas se décomposer en facteurs. Mais cette restriction ne doit pas être considérée comme une objection sérieuse, car il est inutile de se demander si un nombre pair autre que 2 est premier ; pour avoir l'âme en paix, nous avons donc introduit en tête de programme une ligne qui règle cette question.

Liste des variables
(valable pour les programmes 1 et 2)

- N : nombre à étudier
- M : N — 1
- B : base
- E : plus grand exposant de 2 tel que 2^E soit inférieur ou égal à M
- C : 2^E (plus grand nombre de la forme 2^E inférieur ou égal à M)
- D : excédent de M sur C
- R : reste du test de primarité
- F, G, H : variables de calcul en double précision ne dépendant que de N
- S à W : variables de calcul en double précision dépendant de R

N étant ainsi impair, $B^{N-1} - 1$ se décompose au moins en deux facteurs, mais souvent davantage et l'on a : $B^{N-1} - 1 = (B^A - 1)(B^A + 1)(B^{2A} + 1)(B^{4A} + 1) \ldots (B^{(N-1)/2} + 1)$.

L'exposant A est le premier impair obtenu après divisions successives de N — 1 par 2 (A = 85 dans notre exemple). On voit ainsi que le facteur affecté du plus petit exposant (donc impair) nécessite deux tests :

- le reste de B^A/N est-il égal à 1 ?
- le reste de B^A/N est-il égal à N — 1 ?

De plus en plus rares

Dans le programme N — 1 est désigné par M. Le deuxième test devient donc : le reste de B^A/N est-il égal à M ?

Les autres facteurs, jusqu'à $B^{(N-1)/2}$, ne nécessitent qu'un test : le reste de B^K/N est-il égal à M ? L'exposant K prend successivement les valeurs : 2A, 4A, 8A... jusqu'à (N — 1)/2.

Nombres pseudo-premiers restreints inférieurs à 100 000

2 047	49 141
3 277	52 633
4 033	65 281
4 681	74 665
8 321	80 581
15 841	85 489
29 341	88 357
42 799	90 751

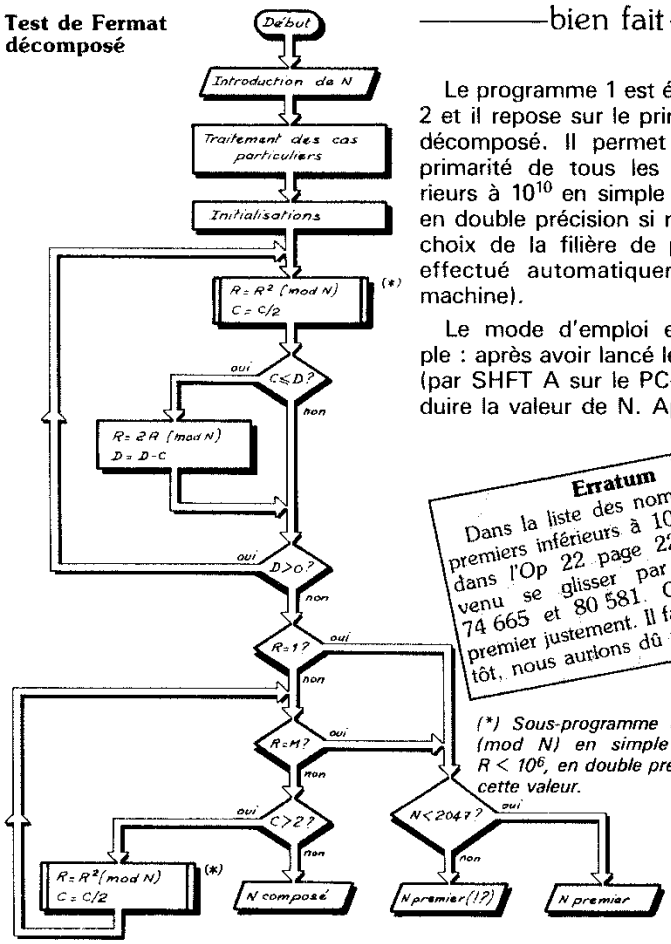
Nous venons donc d'éliminer le nombre 341 de la liste des nombres pseudo-premiers. Mais notre élagage ne s'arrête pas là. En fait, sur les 78 nombres pseudo-premiers en base 2 inférieurs à 100 000 dont nous avons publié la liste dans le précédent numéro de *l'Op* (voir erratum au bas de cette page), il ne subsiste, après passage au test dissocié, que 16 nombres, que nous qualifierons de nombres pseudo-premiers restreints (la liste de ces nombres est reproduite ci-contre). Cela signifie que nous avons éliminé quatre nombres pseudo-premiers sur cinq. La marge d'incertitude du test est donc de 16/100 000, mettons de l'ordre de 2×10^{-4} pour N inférieur à 100 000. Mais en vertu d'une loi de raréfaction des nombres pseudo-premiers, la marge d'incertitude du test décomposé décroît rapidement pour atteindre une fraction de millionième pour des valeurs de N atteignant plusieurs milliards.

Vite fait bien fait

Le programme 1 est établi en base 2 et il repose sur le principe du test décomposé. Il permet d'étudier la primarité de tous les entiers inférieurs à 10^{10} en simple précision, et en double précision si nécessaire (le choix de la filière de précision est effectué automatiquement par la machine).

Le mode d'emploi est très simple : après avoir lancé le programme (par SHFT A sur le PC-1211), introduire la valeur de N. Après un dou-

Test de Fermat décomposé



(*) Sous-programme de calcul de R^2 (mod N) en simple précision pour $R < 10^6$, en double précision au-delà de cette valeur.

Erratum
Dans la liste des nombres pseudo-premiers inférieurs à 100 000, parue dans l'Op 22 page 22, 73 361 est venu se glisser par erreur entre 74 665 et 80 581. Or, 73 361 est premier justement. Il fallait lire ou plutôt, nous aurions dû écrire : 75 361.

Nombres premiers le test de Fermat

LIST
la revue de la programmation individuelle
(voir page 5)

ble bip (sur le PC-1211 seulement), le résultat apparaît sous forme de l'un des trois messages suivants : N COMPOSE, N PREMIER ou N PREMIER (!?).

La présence éventuelle du symbole (!?) signifie que l'assertion « PREMIER » est affectée d'une marge d'incertitude. Notez que ce symbole a été omis pour les nombres inférieurs à 2 047, car en deçà de cette valeur, il n'existe pas de nombre pseudo-premier restreint en base 2, donc pas d'incertitude.

La vitesse d'exécution, loin d'être affectée par la décomposition en facteurs, se trouve même améliorée par rapport aux programmes que nous avons publiés précédemment sur ce thème, puisque le temps d'exécution est inférieur à 30 secondes pour les nombres de 5 chiffres ou moins et de 30 secondes à un peu moins de 2 minutes pour les nombres comportant de 6 à 10 chiffres.

Nous n'avons jusqu'ici évoqué que les tests en base 2. Mais lorsque le test décomposé indique que N est premier ou pseudo-premier en base 2, il est parfaitement légitime d'effectuer un autre test dans une base différente. Si ce deuxième test indique que N est composé, c'est que N était pseudo-premier en base 2, mais qu'en réalité, il est composé. Si le deuxième test indique que N est premier, la probabilité pour qu'il en soit ainsi augmente considérablement. En fait, en base 3 cette probabilité devient une certitude pour tous les nombres inférieurs à 100 000, car nous avons vérifié qu'aucun des 16 nombres pseudo-premiers restreints inférieurs à 100 000 ne résiste à un test décomposé en base 3 (certains d'entre eux : 15 841, 29 341, 49 141, 52 633 et 90 751 résistent cependant à un test simple). En effet, les 16 tests correspondants donnent tous pour résultat : N COMPOSE. Vous pouvez le vérifier en ajustant le programme 1 en base 3 moyennant les modifications suivantes : à la ligne 50, R = 2 devient R = 3 et à la ligne 80, R = 2 * R — N * INT (2* R/N) devient R = 3 * R — N * INT (3 * R/N).

Test absolu de primarité
pour N < 100 000
Programme 2, pour PC-1211
Auteur Pierre Ladislas Gedo
Copyright *l'Ordinateur de poche* et l'auteur

```
500:"B"INPUT " N
    = ";N
510:IF (N>1)*(N<
    E5)*(N=INT N
    )=0BEEP 4:
    GOTO 500
520:IF N=2GOTO 6
    50
530:IF N/2=INT (
    N/2)GOTO 620
540:M=N-1:E=INT
    (LOG (M+.5)/
    LOG 2):B=2
550:C=2^E:D=M-C:
    R=B
560:R=R*R-N*INT
    (R*R/N):C=C/
    2
570:IF C<=DLET R
    =B*R-N*INT (
    B*R/N):D=D-C
580:IF D>0GOTO 5
    60
590:IF R=1GOTO 6
    30
600:IF R=MGOTO 6
    30
610:IF C>2LET R=
    R*R-N*INT (R
    *R/N):C=C/2:
    GOTO 600
620:BEEP 2:PRINT
    N;"COMPOSE":
    END
630:IF N<2047
    GOTO 650
640:IF B=2LET B=
    3:GOTO 550
650:BEEP 2:PRINT
    N;"PREMIER":
    END
```

Les remarques faites pour adapter le programme 1 au PB-100 sont également valables ici. Ce programme peut être introduit dans une autre zone que celle du programme 1. La ligne 510 sera remplacée par deux lignes : 510 IF N > 1 ; IF N < 1 **E** 5 ; IF N = INT N THEN 520 et 515 GOTO 500.

Attention de ne pas confondre les O (quinzième lettre de l'alphabet) et les 0 (zéro). Sur le PC-1211 les zéros ne sont pas "barrés" alors qu'ils le sont sur le PB-100.

Rapide et sûr

Le programme 2, indépendant du programme 1, exploite cette propriété et devient ainsi un test absolu, valable sans restriction dans tous les cas pour les nombres inférieurs à 100 000. L'algorithme utilisé est très simple :

- Tester N en base 2.
  Si la conclusion est : N composé, afficher N COMPOSE.
  Si la conclusion est : N premier (!?), passer au point suivant.
- Tester N en base 3.
  Si la conclusion est : N composé, afficher N COMPOSE.
  Si la conclusion est : N premier (!?), afficher N PREMIER.

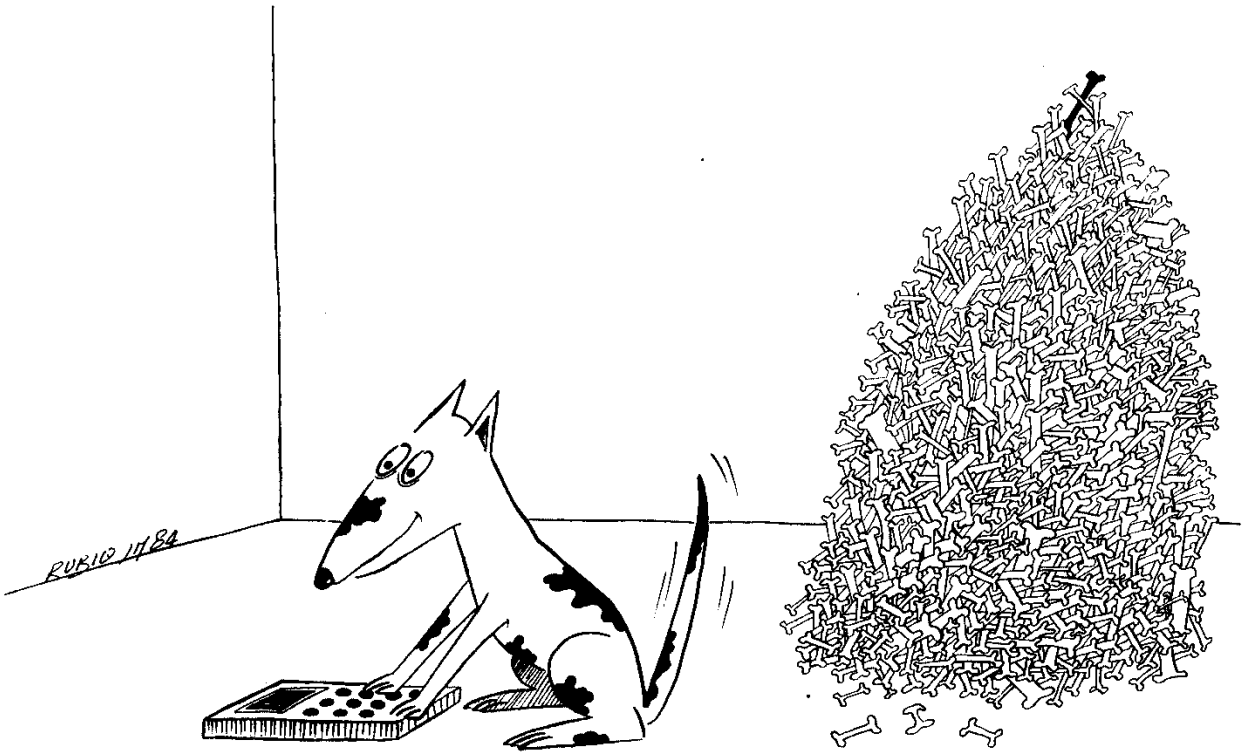
Le mode d'emploi est identique à celui du programme 1, avec initialisation par SHFT B sur le PC-1211.

Le programme tourne rapidement. Les temps d'exécution sont inférieurs à 15 secondes pour tout nombre inférieur à 2 047, à 20 secondes pour tout nombre composé (autre que pseudo-premier en base 2), à 45 secondes pour tout nombre premier ou pseudo-premier en base 2, sur le PC-1211. Le PB-100 est environ cinq fois plus rapide.

Nous avons limité la valeur maximale de N à 100 000 parce que nous avons pu vérifier qu'il n'existe pas de contre-exemple jusqu'à cette valeur, mais il est possible que l'on puisse aller beaucoup plus loin sans risque.

Si vous êtes tenté de rechercher un contre-exemple entre 100 000 et 1 000 000 (il en existe peut-être), transformez **E** 5 en **E** 6 en ligne 510, et armez-vous de patience...

☐ Pierre Ladislas Gedo



Demandez le menu !

Quel morceau du module X-Fonctions désirez-vous croquer ce soir ? Un EMDIR bien arrosé de GETSUB, par exemple ? Faites votre choix.

■ Le programme que je vous propose est bien un menu d'opérations exécutables avec le module X-Fonctions de la HP-41 C. Il est ainsi possible par exemple d'obtenir automatiquement le catalogue des fichiers (fonction EMDIR) en pressant une simple touche du clavier. Mais surtout, le programme apporte des routines utilitaires, tel le tri alphabétique, développées pour faciliter la gestion des fichiers de programmes ou des fichiers ASCII (textes).

L'introduction du programme demande quelques notions de programmation synthétique de la HP-41 pour obtenir des fonctions telles que RCL M (imprimé RCL [) ou STO d. Vous les trouverez notamment dans *l'Op* 19 page 22.

Au tout début, bien sûr, la création

Le programme est assigné à la touche TAN (TFI en mode user). En fait, il s'assigne de lui-même lors de sa première exécution (voir les 4 premières lignes du programme). Une fois lancé, le message OPTION vous demande votre choix. Vous précisez celui-ci en pressant simplement une lettre depuis le clavier. Le « tableau des options du Menu » vous montre quelle lettre provoque quelle action.

L'option C de création d'un fichier de texte vous demande l'introduction de deux paramètres : le nom du fichier (FICHIER ?) et le nombre de caractères que l'on souhaite y stocker (REG). Celui-ci est exprimé en nombre de registres, sachant qu'un registre permet de conserver 7 caractères. Enfin, le message TEXTE vous invite à taper votre texte. Une phrase (ou enregistrement) est terminée par R/S. On peut alors passer à l'enregistrement suivant. Si rien n'est tapé, cela termine la procédure d'entrée de texte. Toutes ces phrases, ces enregistrements, forment le fichier de texte.

L'option D comme Destruction

Demandez le menu ! HP-41 + X-fonctions

Dès à présent, écrivez à
LIST
5 place du Colonel Fabien,
75491 PARIS CEDEX 10

sert à effacer un enregistrement. Presser D fait apparaître FICHIER ? ; comme précédemment, répondre par le nom du fichier à traiter, ou simplement R/S pour le fichier en cours. Chaque enregistrement est visualisé pendant quelques secondes. Presser D pour commander sa destruction, R/S, ou attendre, pour passer à la phrase suivante.

I permet d'ajouter un enregistrement au fichier. Après avoir répondu au traditionnel FICHIER ?, introduisez votre texte, tout simplement.

L'option L est une visualisation rapide de toutes les phrases d'un fichier de texte.

Avec M vous pouvez modifier un enregistrement. Choisissez votre fichier, puis, lorsqu'apparaît la chaîne à changer, inscrivez le nouveau texte, directement.

L'option R est particulièrement intéressante pour traiter les informations du fichier. Il s'agit d'une recherche d'enregistrement effectuée à partir d'un ou plusieurs caractères de la phrase à retrouver. Ceci est utile si par exemple votre fichier n'est autre qu'un carnet d'adresses. Le mot recherché peut très bien se situer n'importe où dans une

Demandez le menu !
Programme pour HP-41 C et module X-fonctions
Auteur Robert Séraline
Copyright *l'Ordinateur de poche* et l'auteur

```
01♦LBL "FI"
02 "OFI"
03 ATOX
04 PASN
05♦LBL 10
06 "♦♦αα♦"
07 RCL I
08 STO d
09 "OPTION"
10 AVIEW
11 RCL b
12 PSE
13 FC?C 23
14 STO b
15 ATOX
16 GTO IND X
17 FC? 25
18 GTO 10
19♦LBL 67
20 SF 01
21 XEQ 05
22 "REG"
23 AOFF
24 PROMPT
25 CLA
26 ARCL 01
27 CRFLAS
28 FC? 25
29 GTO 10
30 GTO 07
31♦LBL 73
32 SF 02
33 GTO 68
34♦LBL 77
35 SF 04
36 GTO 68
37♦LBL 76
38 SF 00
39♦LBL 68
40 XEQ 05
41 FS? 02
42 GTO 07
43 RCL b
44 SF 25
45 GETREC
46 FC? 25
47 GTO 10
48 AVIEW
49 FS? 00
50 STO b
51 FS? 04
52 GTO 04
53 14
54 GETKEY
55 X=Y?
56 DELREC
57 RCL Z
58 STO b
59♦LBL 04
60 34
61 TONE 2
62 GETKEY
63 CF 23
64 X=Y?
65 PROMPT
66 RCL Z
67 FC?C 23
68 STO b
69 DELREC
70 INSREC
71 STO b
72♦LBL 07
73 RCL b
74 SF 25
75 CF 23
76 "TEXTE"
77 AON
78 PROMPT
79 FC?C 23
80 GTO 10
81 APPREC
82 FC? 25
83 GTO 10
84 STO b
85♦LBL 88
86 SF 04
87 GTO 83
88♦LBL 71
89 SF 03
90♦LBL 83
91 "PRGM"
92 PROMPT
93 ASTO X
94 FS? 03
95 GETSUB
96 SAVEP
97 FS? 03
98 GTO IND X
99 FS? 04
100 PCLPS
101 GTO 10
102♦LBL 82
103 XEQ 05
104♦LBL 09
105 XEQ 06
106 "CHAINE"
107 AON
108 PROMPT
109 FC?C 23
110 GTO 10
111 ASTO 03
112 RCL b
113 CLA
114 ARCL 03
115 POSFL
116 X<0?
117 GTO 09
118 RCLPT
119 INT
120 SEEKPT
121 GETREC
122 RCL Z
123 PROMPT
124 STO b
125♦LBL 80
126 EMDIR
127 RCL b
128 SF 25
129 SF 06
130 XEQ 05
131 PURFL
132 FC? 25
133 GTO 10
134 RDN
135 STO b
136♦LBL 05
137 RCL b
138 "FICHIER?"
139 PROMPT
140 FS? 06
141 RTN
142 FS?C 23
143 ASTO 01
144 FS? 01
145 RTN
146♦LBL 06
147 SF 25
148 CLA
149 ARCL 01
150 ,
151 SEEKPTA
152 RDN
153 FS? 25
154 RTN
155 FS? 07
156 GTO 10
157 TONE 5
158 SF 07
159 STO b
160♦LBL 69
161 EMDIR
162 VIEW X
163 PSE
164 GTO 10
165♦LBL 75
166 XEQ 05
167 *
168 APPREC
169 ATOX
170 STO 03
171 65,09
172 STO 00
173 ,
174 SEEKPT
175♦LBL 01
176 SF 25
177 RCL 03
178 GETREC
179 FC? 25
180 GTO 03
181 RCLPT
182 INT
183 STO 04
184 RDN
185 ATOX
186 X=Y?
187 GTO 03
188 RCL 00
189 INT
190 X=Y?
191 GTO 02
192 GTO 01
193♦LBL 03
194 ,
195 SEEKPT
196 ISG 00
197 GTO 01
198 GTO 04
199♦LBL 02
200 RCL 04
201 SEEKPT
202 GETREC
203 DELREC
204 APPREC
205 ,
206 SEEKPT
207 GTO 01
208♦LBL 04
209 ,
210 SEEKPT
211 DELREC
212♦LBL 70
213 OFF
214 END
```

Tableau des options du Menu

C - création d'un fichier de texte
D - destruction d'un enregistrement
I - ajout d'un enregistrement
K - classement alphabétique d'un fichier de texte
L - visualisation rapide d'un fichier de texte
M - modification d'un enregistrement
R - recherche d'un enregistrement
E - catalogue général des fichiers
G - rappel en mémoire centrale d'un programme
P - destruction d'un programme en X-mémoire
S - sauvegarde d'un programme vers l'X-mémoire
F - fin des travaux et extinction de la HP-41.

phrase. Toutes les occurrences sont retrouvées, c'est-à-dire que la même chaîne située à différents endroits du fichier sera successivement affichée.

L'option K réalise le tri alphabétique du fichier. Les phrases sont réordonnées dans l'ordre croissant, il suffit d'indiquer le nom du fichier à classer et d'attendre, parfois assez longtemps. C'est pourquoi il est recommandé de trier les longs fichiers... la nuit ! La HP-41 C s'éteindra toute seule une fois le tri terminé.

Cette extinction automatique de la HP est une des options du menu, l'option F. Ce peut être un moyen de sortir du programme.

Enfin, la lettre E correspond au catalogue de tous les fichiers, textes comme programmes. Ces derniers sont gérés à l'aide de trois options : G, P et S, qui correspondent respectivement à GETSUB, PURFL et SAVEP. Dans l'ordre, ces fonctions réalisent le rappel en mémoire centrale d'un programme conservé dans la mémoire auxiliaire du module X-Fonctions, la destruction d'un programme en X-mémoire et le transfert d'un programme de la mémoire centrale vers le module.

Quand on agite tous les drapeaux

De nombreuses instructions synthétiques élémentaires sont employées. Le plus souvent elles permettent une manipulation rapide de *tous* les drapeaux du système (les *flags* ou indicateurs binaires).

Les lignes 02 à 04 du programme assignent automatiquement le label ᵀFl à la touche TAN. Ce qui apparaît comme un O au pas 02 de la liste est un caractère visualisé ⌘ de code 25. Ces pas synthétiques peuvent être remplacés par ᵀFl, 25, PASN.

Les lignes 05 à 18 initialisent les drapeaux à l'aide d'une chaîne de caractères spéciaux dont les codes sont stockés dans d, le registre des drapeaux. Les codes de ces caractères mettent à zéro (CF) tous les indicateurs sauf les nᵒˢ 21, 25, 26 et 27 (qui peuvent aussi être armés normalement avec l'instruction SF) ainsi que les nᵒˢ 31, 40, 48 (AON) et 55 (imprimante). Les codes de la chaîne synthétique sont les suivants : 0, 0, 4, 113, 0, 128 et 129.

Pour ceux qui ne sauraient programmer une telle chaîne (fonction mu), faire simplement ᵀ (deux espaces), 4 XTOA, 113 XTOA, 0 XTOA, 128 XTOA, 129 XTOA. Le reste : RCL M, STO d est inchangé ; conclure enfin par CF 02.

Vous aurez certainement noté qu'à chaque lettre d'option correspond un sous-programme dont le label n'est autre que le numéro du code ASCII de la lettre. Ainsi à la lettre A correspondrait un LBL 65. En continuant sur le même principe, vous pourrez rajouter au programme autant d'options que vous le souhaitez.

☐ Robert Séraline

A vous de jouer sur le PB-100

Petit, rapide et doué pour les maths, le PB-100 permet aussi de se divertir. Parmi les programmes de jeux que nous avons reçus pour cet ordinateur, nous en avons sélectionné cinq. Ils présentent l'avantage de fonctionner sans imprimante et d'être assez courts pour convenir au PB-100 dans sa version de base. Aucun de ces programmes ne dépasse donc la barrière fatidique des 544 octets...

La mémoire des signes

■ Cinq mots bizarres se présentent l'un à la suite de l'autre à l'écran : ce sont des « mots de signes » (1). Ils sont bien difficiles à lire et plus difficiles encore à retenir. Ils s'appellent respectivement L1, L2, L3, L4 et L5. Un sixième « mot de signes » apparaît alors. Il s'appelle Te, comme Test. Si Te est semblable à l'un des mots L1, L2, L3, L4 ou L5, frapper alors O (comme Oui, Te a déjà été vu). Dans le cas contraire, frapper N (comme Non, Te n'a pas été vu, il est nouveau).

Si votre mémoire ne vous a pas trahi, vous bénéficiez d'un message de félicitations. Sinon, un message signale votre erreur. Le nombre d'essais est introduit en début de partie, et le score final est affiché sous la forme d'une note sur 20.

□ André Turlure

(1) Le programme d'André Flédrich pour HP-41 C, dans l'Op *7 page 41, proposait un jeu semblable.*

Test mémoire
Programme pour PB-100
Auteur André Turlure
Copyright *l'Ordinateur de poche* et l'auteur

```
438 pas

 10 VAC :PRINT "Tes
    t Memoire"
 20 INPUT "Nb essai
    s",M
 30 FOR L=1 TO M
 40 PRINT :PRINT "u
    n instant";
 50 $="Σ%()↑≠≤-+*/=
    ?#≥$&Ωμπ"
 60 A$="":B$="":C$=
    "":D$="":E$="":
    F$="":J$="N"
 70 FOR Q=1 TO 6:FO
    R P=0 TO 5
 80 R=INT (RAN#*20)
    +1:K$=MID(R,1)
 90 A$(P)=A$(P)+K$:
    NEXT P:NEXT Q
100 FOR P=0 TO 4
110 PRINT :PRINT P+
    1;CSR 0;"L";CSR
     5;A$(P);
120 GOSUB 500:NEXT
    P
140 PRINT :FOR Q=0
    TO 59:NEXT Q
150 R=INT (RAN#*10)
    :IF R>4;R=5:J$=
    "O"
160 PRINT "Te";CSR
    5;A$(R);
170 IF KEY="" THEN
    170
180 PRINT
200 IF KEY=J$;PRINT
     CSR 3;"'erreur
    '";:GOSUB 500:G
    OTO 250
210 PRINT CSR 3;"Br
    avo !";:N=N+1:G
    OSUB 500
250 NEXT L
270 PRINT :PRINT "S
    CORE:";INT (20*
    N/M);"/20"
300 END
500 FOR Q=0 TO 139:
    NEXT Q:RETURN
```

Un atterrissage en douceur

■ Même sans songer à piloter un avion de tourisme, il peut être amusant de tester ses capacités d'atterrissage sur un PB-100. C'est-à-dire d'essayer de poser un appareil en douceur. Au moins ne risque-t-on pas de « casser du bois ».

Deux nombres défilent à droite et à gauche de l'écran. Ils symbolisent l'altitude de l'avion. Ils partent de 200 et tendent à se rapprocher de zéro, mais ils ne vont pas à la même vitesse. Le but du jeu est d'obtenir qu'ils atteignent zéro au même moment. Et ceci grâce à deux touches du clavier : « 1 » qui ajoute dix au nombre de gauche et « 3 » qui ajoute dix au nombre de droite.

Si l'avion se pose au moment de l'égalité à zéro, vous aurez gagné des points de « bonus ». Avez-vous heurté la piste du bout de l'aile droite ou gauche ? Ils ne vous reste plus qu'à recommencer : vous ferez sans doute mieux une prochaine fois.

☐ Jean-Pierre Raby

Atterrissage en douceur
Programme pour PB-100
Auteur Jean-Pierre Raby
Copyright *l'Ordinateur de poche* et l'auteur

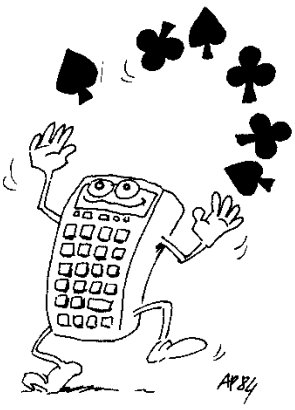
```
440 Pas

P1
  5 VAC
  6 N=200
  7 PRINT "Pilote";
    "  ";"bonus:200
    Pts"
  8 A=200:B=A:F=0:E
    =0:D=0:P=0
  9 L=L+1
 12 C=INT (RAND*2)+
    1:D=INT (RAND*2
    )+1
 13 IF C=D THEN 12
 20 A=A-C:B=B-D
 22 F=A+P:E=B+D:GOS
    UB #2
 30 Y$=KEY
 31 IF Y$="1";P=P+1
    0:GOSUB #3
 32 IF Y$="3";D=D+1
    0:GOSUB #3
 34 Z=0
 36 IF F=Z:IF E≠Z:P
    RINT "HEURTE A
    DROITE":GOTO 40
 37 IF F≠Z:IF E=Z:P
    RINT "HEURTE A
    GAUCHE":GOTO 40
 38 IF F=Z:IF E=Z:P
    RINT "ATTERRI!"
    :GOTO 42
 39 GOTO 20
 40 N=N-30:PRINT "R
    ESTE":N;" EN";L
    ;"TOUR":GOTO 60
 42 N=N+30:PRINT "G
    AINS":N;" EN";L
    ;"TOUR"
 60 PRINT "EXE pour
     repartir":GOTO
     8

P2
 10 PRINT CSR 0;F;C
    SR 7;E;:PRINT :
    RETURN

P3
 10 PRINT CSR 0;P;C
    SR 7;D;:RETURN
```

Ce programme occupe trois zones qui sont ici P1, P2 et P3.

Récréation en piques et en trèfles

■ Trois « piques » et trois « trèfles » se trouvent face à face. Tout semble bien se passer et pourtant, ils ne seront vraiment à l'aise que lorsqu'ils auront inversé leurs positions : les piques à la place des trèfles et les trèfles à la place des piques.

Pour cela, il suffit d'introduire un chiffre compris entre 0 et 8. Ce chiffre représente la position de la figure à déplacer. Mais les déplacements ne se font pas tous de la même façon. Aussi faut-il commencer par découvrir ce qui se passe. Si le jeu



A vous de jouer sur le PB-100

Piques et trèfles
Programme pour PB-100
Auteur André Turlure
Copyright *l'Ordinateur de poche* et l'auteur

```
422 pas

 10 FOR J=0 TO 8 ST
    EP 4:A$(J)="_":
    NEXT J
 20 FOR J=1 TO 3:A$
    (J)="♠":NEXT J
 30 FOR J=5 TO 7:A$
    (J)="♣":NEXT J
 40 FOR J=0 TO 8:PR
    INT CSR J;A$(J)
    ;
 50 NEXT J
 60 IF A$+B$+C$+D$+
    E$+F$+G$+H$+I$+
    J$="_♣♣♣_♠♠♠_"
    THEN 160
 70 INPUT K:L$=A$(K
    ):IF L$="_" THE
    N 40
 80 IF L$="♣" THEN
    120
 90 IF A$(K+1)="_";
    A$(K+1)="♠":A$(
    K)="_":GOTO 40
100 IF A$(K+2)="_";
    A$(K+2)="♠":A$(
    K)="_":GOTO 40
120 IF L$*"♣" THEN
    40
125 IF K=0 THEN 40
130 IF A$(K-1)="_";
    A$(K-1)="♣":A$(
    K)="_":GOTO 40
140 IF A$(K-2)="_";
    A$(K-2)="♣":A$(
    K)="_":GOTO 40
150 GOTO 40
160 FOR J=0 TO 99:N
    EXT J
170 PRINT :PRINT "R
    EUSSI !"
```

est bloqué, on le relance par SHIFT n, le programme étant en zone Pn.

Ce petit casse-tête admet au moins une solution. Elle peut être trouvée assez rapidement...

☐ André Turlure

Tir à l'arc

■ Il vous faudra faire preuve de beaucoup d'attention pour devenir un champion de ce tir à l'arc un petit peu particulier. Il réclame, en effet, de la rapidité et de la précision. De plus, l'affichage est assez pâle, même avec le contraste poussé au maximum (ce peut être l'occasion de changer vos piles !).

Vous vous trouvez à gauche de l'écran, symbolisé par Σ, et vous visez une espèce de gibier assez rare, symbolisée par un point ou une apostrophe selon sa position. Ce gibier se déplace vite, de bas en haut et de la droite vers la gauche de l'écran. Il ne peut être touché que s'il se trouve à mi-hauteur sur l'affichage (ni en haut, ni en bas).

Pour décocher vos flèches, vous disposez de la touche **E**, en bas et à

Tir au vol
Programme pour PB-100
Auteur Laurent Boulanikian
Copyright *l'Ordinateur de poche* et l'auteur

```
494 pas

 10 C$=".":D$="·":E
    $="'":K=3:N=6:M
    =0:H=4:MODE H:R
    =0
 20 F=11:J=2
 30 FOR I=1 TO N:B$
    =KEY:IF B$="E"
    THEN 80
 35 NEXT I
 40 J=J+1:IF J=5;J=
    2
 45 F=F-1/K:IF F<0;
    F=0
 50 PRINT :PRINT "Σ
    ";CSR F;A$(J);:
    IF INT F=0 THEN
     150
 60 GOTO 30
 80 A=1:R=R+1:IF R>
    3 THEN 40
 90 PRINT :PRINT "Σ
    ";CSR F;A$(J);C
    SR A;"→";
 95 IF A=INT F;IF J
    =3 THEN 120
100 A=A+1;IF A=12 T
    HEN 30
110 GOTO 90
120 PRINT :PRINT CS
    R 2;"TOUCHE";:F
    OR I=1 TO ((9-N
    )*F-H)/(.3+K)
125 M=M+1:PRINT
130 PRINT CSR 3;M;:
    NEXT I:N=N-1:IF
     N=-1;N=0:K=K-1
    /5
140 IF K≤0.4;K=0.4
145 R=0:GOTO 20
150 FOR I=1 TO 5:FO
    R F=1 TO 3:PRIN
    T
155 PRINT D$;:NEXT
    F:FOR F=1 TO 3
160 PRINT CSR 0;"Σ"
    ;:NEXT F:NEXT I
    :PRINT
165 PRINT "SCORE";M
    :H=H+1
170 IF H>6 THEN 300
180 MODE H:R=0:GOTO
     20
300 IF M>Z;Z=M
310 PRINT CSR 0;"Me
    illeur score";Z
    :GOTO 10
```

droite du clavier alphabétique. Cette touche ne peut servir que trois fois par tour à moins que vous ne fassiez mouche. Dans ce cas, vous récupérez trois flèches. La proie, elle, se défend en augmentant sa vitesse. Dès qu'elle vous a rejoint, vous « clignotez ». Le tour suivant va être engagé.

Une partie se compose de trois tours qui se distinguent l'un de l'autre par l'indicateur angulaire allumé : DEG, RAD ou GRA. A la fin, le score est affiché, ainsi que celui de la meilleure partie. Ce dernier disparaîtra avec l'instruction VAC qui peut être présente dans un autre programme.

☐ Laurent Boulanikian

Une revue pour
les amateurs de programmation :
LIST
(voir page 5)

Croq' tout
Programme pour PB-100
Auteurs Luis Godinho et Jean-Marc Odobez
Copyright *l'Ordinateur de poche* et les auteurs

```
535 PAS

  2 PRINT "CROQ'TOU
    T !":GOSUB 99
  4 A=0:M=0:I=11:L=
    0:D=0
  6 FOR B=11 TO 0 S
    TEP -1
  8 PRINT CSR A;"o"
    ;CSR B;">";
 10 IF B≥I;PRINT CS
    R I;"x"::IF A=I
     THEN 60
 12 GOSUB 90:PRINT
    CSR B;"-"::GOSU
    B 90:IF A=B THE
    N 72
 14 IF D>7 THEN 20
 16 IF A=F THEN 50
 18 PRINT CSR F;"$"
    ;
 20 D=D+1:IF D=30:D
    =0:F=INT (RAND*
    12)
 22 G$=KEY:IF G$="P
    ";A=A+1:M=M+1:I
    F A=B THEN 70
 24 IF G$="Q":A=A-1
    :M=M-1
 26 IF B=0:B=12:H=I
    NT (RAND*12)
 28 IF B=H;I=INT (R
    AND*12)
 30 IF A<0:A=11
 32 IF A>11:A=0
 34 PRINT :NEXT B
 50 E=INT (RAND*31)
    :M=M+E:D=8
 52 J=F:IF F>5;J=F-
    3
 54 PRINT CSR J;E;:
    GOSUB 90:GOTO 2
    0
 60 I=9:IF A≥5;I=3
 62 GOTO 74
 70 PRINT CSR A-1;"
     ";
 72 IF G$="W";M=M+5
    :GOTO 26
 74 GOSUB 99:L=L+1:
    IF L≥3;GOTO 78
 76 PRINT " MORT N°
    ";L;:GOSUB 99:G
    OTO 22
 78 PRINT "FINI !";
    :GOSUB 99
 80 PRINT "SCORE=";
    M:GOTO 2
 90 FOR Z=0 TO 8:NE
    XT Z:RETURN
 99 FOR C=0 TO 200:
    NEXT C:PRINT :R
    ETURN
```

Croq' tout

■ Le « croq' tout » est un petit animal qui dévore tout sur son passage. Comme il emprunte toujours le même chemin et que sa vitesse est constante, il est possible de l'éviter. Il fait la traversée de l'écran de la droite vers la gauche et arrivé au bout, il réapparaît tout à coup à la droite et poursuit son chemin.

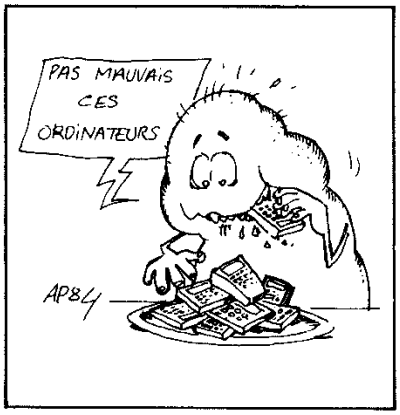
Vous êtes symbolisé par un petit rond positionné, au départ, à gauche de l'écran et vous voulez absolument éviter le croq' tout. Vous avez la possibilité d'aller à droite grâce à la touche P (vous marquez alors un point), d'aller à gauche grâce à la touche Q (vous perdez alors un point) ou de disparaître de la vue du croq' tout lors de son passage sur vous grâce à la touche W (auquel cas vous marquez cinq points).

A tout moment, une croix peut venir s'interposer sur une position quelconque de l'écran et pendant une durée variable. Si elle vous barre le passage, la seule solution consiste à attendre que le croq' tout l'ait dévorée. Attention à ne pas la heurter : elle vous ferait disparaître. Si par malchance elle surgit directement sur votre position, il n'y a rien à faire.

Pour vous aider à améliorer votre score, un dollar se manifeste régulièrement à un endroit imprévisible de l'afficheur. Si vous l'atteignez avant qu'il ne s'évanouisse, vous gagnez un nombre de points compris entre 0 et 30. C'est parfois très payant mais il arrive aussi que l'on ne gagne que deux points tout en en perdant cinq pour atteindre le dollar en reculant...

Une partie se joue sur trois tours correspondant chacun à une vie. Si vous voulez vous amuser, ne vous laissez pas dévorer trop vite.

Luis Godinho
☐ Jean-Marc Odobez



Les grenouilles n'ont peur de rien

Six grenouilles, parfaitement intrépides, se lancent dans une aventure moderne et très périlleuse : la traversée d'une autoroute. Votre tâche consiste à les guider pour leur éviter tout accident (programme pour ZX 81).

■ Les Britanniques, dit-on, n'aimeraient pas les grenouilles. En fait, ils ne comprennent pas que l'on puisse les manger. On aime donc les grenouilles, Outre-Manche, mais trop (ou pas assez) pour en faire un plat. Toujours est-il qu'il nous vient de là-bas un jeu vidéo, devenu un classique du genre, où l'on doit aider des grenouilles à traverser une autoroute. C'est une version de ce jeu, adaptée au ZX 81 (16 Ko) qui vous est proposée ici.

Le programme dans ses grandes lignes

5-90 : initialisation des variables A$ (voitures)
100-129 : plan du jeu
130-135 : initialisation de la position des grenouilles
140-190 : déplacement des voitures
200-270 : sous-programme de déplacement de la grenouille
300-320 : cas où l'on a réussi une traversée
325-360 : on a gagné la partie
370-385 : cas où l'on a raté une traversée
390-410 : on a perdu.

Variables utilisées

A$(1), A$(2), A$(3) : les voitures
D : niveau de difficulté
G : nombre d'erreurs encore autorisées
E : nombre de traversées réussies
H : nombre aléatoire valant entre 0 et 13
P1 et P : position de la grenouille (colonne et rangée)
W : examen de l'écran à la position P1, P
F et Q : variables de boucles.

Le niveau de difficulté varie, au choix du joueur, entre 1 (très difficile) et 12 (facile). Il est demandé dès que le programme est lancé. Le plan du jeu apparaît ensuite à l'écran. Un « M » inscrit sur fond noir représente, en bas et à gauche, l'endroit d'où partent les grenouilles. Chacune d'entre elles va tout droit, sans prêter garde à la circulation. Heureusement pour elle, vous êtes là : en appuyant respectivement sur les touches 5 et 8, vous la guiderez vers la gauche ou la droite.

La victoire est acquise lorsque six grenouilles ont rejoint, de l'autre côté de l'autoroute, six petites cases qui apparaissent en grisé, et vous n'avez droit qu'à trois « accidents ». Le temps n'intervenant pas, aucun score n'est tenu, et c'est un affichage particulier qui vous indiquera si vous avez gagné ou perdu. La compétition consiste donc ici à choisir des niveaux de difficulté de plus en plus ardus.

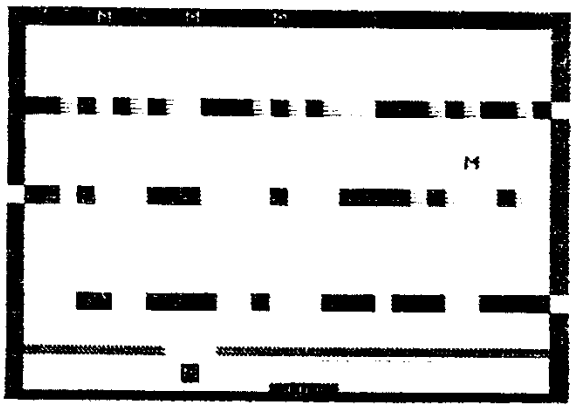
Ah, j'oubliais : si vous le voulez, rien ne vous interdit d'imaginer que les grenouilles sont en réalité des hérissons...

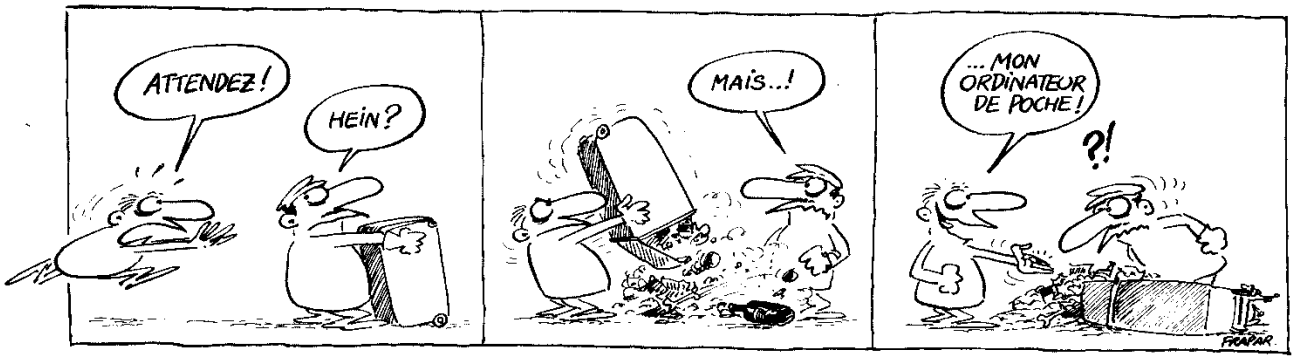
☐ François Fayard

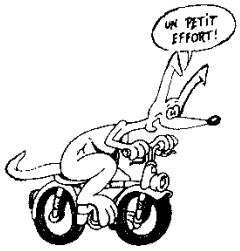
Six petites grenouilles
Programme pour ZX 81 (+ 16 Ko)
Auteur François Fayard
Copyright *l'Ordinateur de poche* et l'auteur

```
   5 CLS
  10 DIM A$(3,160)
  11 PRINT AT 0,0;"DIFFICULTE ?(
1 TRES DUR--12)"
  12 INPUT D
  13 IF D<1 OR D>12 THEN GOTO 12
  14 CLS
  15 FAST
  16 LET G=2
  17 LET E=0
  20 FOR F=1 TO 3
  30 FOR Q=32 TO 128
  40 LET H=INT (RND*14)
  50 LET A=0
  60 IF H>D THEN LET A=128
  70 LET A$(F,Q)=CHR$ A
  80 NEXT Q
  90 NEXT F
 100 PRINT AT 0,0;"
 111 FOR F=1 TO 20
 112 PRINT TAB 0;"■";TAB 31;"■"
 114 NEXT F
 120 PRINT AT 21,0;"
 125 PRINT AT 19,0;"
 129 SLOW
 130 LET P1=9
 135 LET P=20
 136 PRINT AT P,10;"■"
 140 FOR F=1 TO LEN A$(1)-31
 145 PRINT AT P,P1;" "
 150 PRINT AT 16,0;A$(1,F TO F+3
1)
 160 PRINT AT 5,0;A$(2,F TO F+31
)
 170 PRINT AT 10,0;A$(3,LEN A$(3
)-30-F TO LEN A$(3)-F-1)
 180 IF F>20 THEN GOSUB 200
 185 NEXT F
 190 GOTO 140
 200 LET P=P-.5
 210 LET P1=P1+(INKEY$="8")-(INK
EY$="5")
 220 PRINT AT P,P1;
 230 LET W=PEEK (PEEK 16398+256*
PEEK 16399)
 235 PRINT AT P,P1;"M"
 240 IF W=128 OR W=178 THEN GOTO
 370
 260 IF W=136 THEN GOTO 300
 270 RETURN
 300 PRINT AT 0,P1;"■"
 310 LET E=E+1
 320 IF E<>6 THEN GOTO 130
 325 FOR F=1 TO 10
 330 PRINT AT 0,0;"
 331 FOR Q=1 TO 5
 332 NEXT Q
 335 PRINT AT 0,0;"
 336 FOR Q=1 TO 3
 337 NEXT Q
 340 NEXT F
 350 PRINT AT 10,0;"BRAVO,VOUS A
VEZ GAGNE...
 355 PAUSE 4E4
 360 RUN
 370 LET G=G-1
 375 PRINT AT P,P1;"■"
 380 PRINT AT 21,16+G;"■"
 385 IF G>-1 THEN GOTO 130
 390 PRINT AT 10,0;"VOUS AVEZ FA
IT ECRASER TOUTES VOS GRENOUILLE
S..."
 400 PRINT ,,"PAR CONSEQUENT,VOU
S AVEZ PERDU"
 405 PAUSE 4E4
 410 RUN
 500 SAVE "FROGGE■"
```

EXEMPLE D"EXECUTION



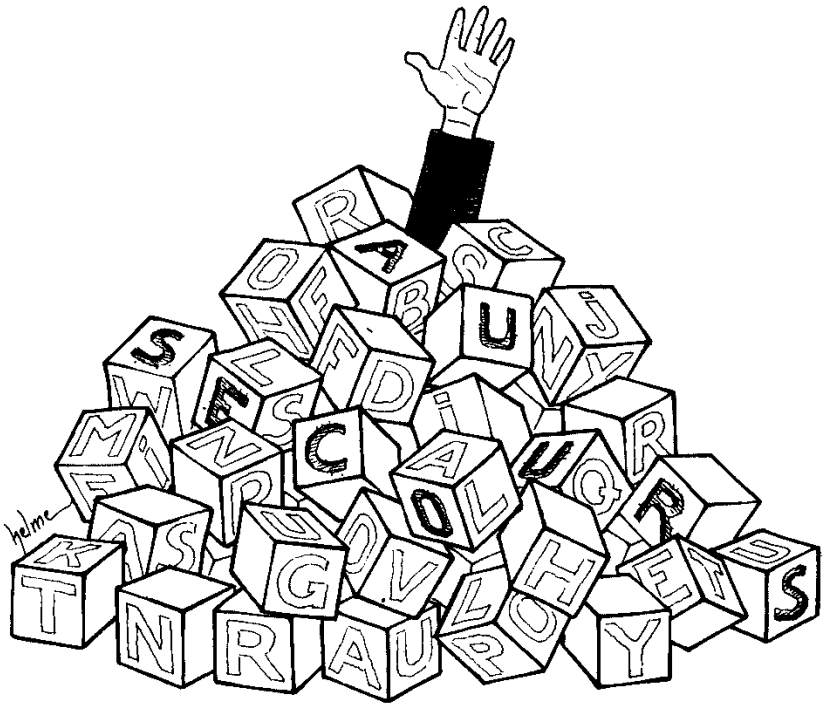




Dix mots jouent à cache-cache

(PC-1500 et CE-150)

Il en est qui perdent leur latin, d'autres leur tête ou leur chemin. Nous, avec ce programme et un Sharp PC-1500 nous perdons tout cela en même temps : allez donc retrouver vos mots dans un tel fouillis !



```
PLXBBDIRAREOYYMWXD
SGQLCJONMIMSCQIDSU
VKROBSEDANRMUCNWUE
AUZRHFXEJEGUMSDLUG
JNANCTXPDZRGTIGROD
LELBORXCCEJMJOAVGA
FONFIANPCEVIKMHQZJ
JGIDNEGLMVERIAPVBL
ETFEEQQTOLBTCHJMXB
LOIVFWSFVHBSJBHIJA

NOIR
MARINE
ROUGE
CHAMOIS
VERT
VIOLET
INDIGO
ORANGE
BLANC
JAUNE
```

Un exemple d'exécution : la grille de cent lettres où le programme a caché les dix mots reproduits ci-dessus.

■ Pas très sérieux, l'ordinateur ! On lui confie dix mots, ni plus, ni moins, en lui demandant de les conserver gentiment dans sa mémoire. Et toc ! Impossible de remettre la main dessus : il les a complètement mélangés. Et pourtant, ils sont là ces mots, tous, à peine déformés d'ailleurs. Quand on croit en avoir retrouvé un, il s'évanouit tout à coup... Et dire que le « jeu » consiste à retrouver dans le mélange de lettres ces dix mots déformés par l'ordinateur !

RUN, et le PC-1500 demande les 10 mots puis imprime la matrice du puzzle : « débrouillez-vous ! ».

Toutes les lettres d'un même mot sont jointes les unes aux autres, mais tel un serpent, le mot s'étire dans les quatre directions verticales et horizontales (Sud, Nord, Est et Ouest). Heureusement, les lettres se touchent toujours, en cas de changement de direction, à angle droit.

On peut aussi jouer à deux

Vous devez trouver la solution tout seul, mais si vous voulez donner votre langue au chat, DEF S vous l'imprimera.

A propos de

LIST

avez-vous lu notre éditorial en page 5 ?

```
SOLUTION :

------IRAR--------
------ONMI--------
---------N-------E
AU-------E---S--UG
JNANC--------I-RO-
-ELBOR-------O----
-O--IAN------M----
-GIDNEG--VER-A----
ET---------TCH----
LOIV--------------
```

```
SOLUTION :

------IRAR--------
------ONMI--------
---------N-------E
AU-------E---S--UG
JNANC--------I-RO-
-ELBOR-------O----
-O--IAN------M----
-GIDNEG--VER-A----
ET---------TCH----
LOIV--------------
```

Voilà quelle était la solution. Vous l'aviez trouvée ? Bravo !

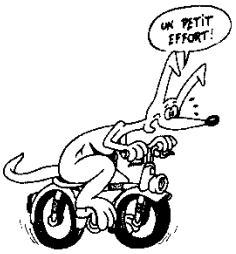
Il ne reste plus qu'à vérifier que l'ordinateur n'a pas triché !

Plus fort encore (si le cœur vous en dit), supprimez LPRINT M$ (A) à la ligne 15 et faites introduire les mots par une autre personne. Si vous les retrouvez avec le titre comme seule indication : chapeau !

☐ Alain Pechmajou

Puzzle mot
Programme pour PC-1500 et CE-150
Auteur Alain Pechmajou
Copyright *l'Ordinateur de poche* et l'auteur

```
  1:WAIT 0:PAUSE "
    * PUZZLE MOT *
 10:CLEAR :DIM C$(
    10,18)*1,V$(0)
    *64,S$(10,18)*
    1,M$(10):INPUT
    " TITRE ?";T$:
    CSIZE 2:
    LCURSOR 4
 11:LPRINT T$:LF 2
 12:REM CHOIX DES
    MOTS
 13:FOR A=1TO 10:
    BEEP 1,200:
    CSIZE 1:
    LCURSOR 15
 15:CLS :PRINT "MO
    T No ";A;" ";:
    INPUT M$(A):L=
    LEN M$(A):
    LPRINT M$(A)
 20:RANDOM :X=RND
    10:Y=RND 18:V$
    (0)="":IF C$(X
    ,Y)<>""GOTO 20
 22:FOR I=1TO L
 24:X$=STR$ X:IF X
    <10LET X$="0"+
    STR$ X
 26:Y$=STR$ Y:IF Y
    <10LET Y$="0"+
    STR$ Y
 27:V$(0)=V$(0)+X$
    +Y$:E=0
 30:C$(X,Y)=MID$ (
    M$(A),I,1):S$(
    X,Y)=C$(X,Y):N
    =X:M=Y
 32:X=N:Y=M
 33:REM DEPLACEMEN
    T ALEATOIRE
 35:IF RND 0>.5LET
    V=1:GOTO 45
 40:V=-1
 45:IF RND 0>.5LET
    X=X+V:GOTO 55
 50:Y=Y+V

 55:IF X>10OR Y>18
    OR X<1OR Y<1
    GOTO 200
 60:IF C$(X,Y)<>""
    GOTO 200
 65:NEXT I:NEXT A
 67:REM MARQUAGE C
    ASES VIDES
 70:FOR X=1TO 10:
    FOR Y=1TO 18:
    IF C$(X,Y)=""
    LET C$(X,Y)=
    CHR$ (RND 26+6
    4):S$(X,Y)="-
 72:NEXT Y:NEXT X:
    CSIZE 2:LF 2
 75:REM DESSIN GRI
    LLE
 80:FOR X=1TO 10:
    FOR Y=1TO 18:
    LPRINT C$(X,Y)
    ;:NEXT Y:
    LPRINT "":NEXT
    X:LF 3:END
 85:"S"LPRINT "SOL
    UTION :":LF 1
 90:FOR X=1TO 10:
    FOR Y=1TO 18:
    LPRINT S$(X,Y)
    ;:NEXT Y:
    LPRINT "":NEXT
    X:LF 3
100:END
200:E=E+1:IF E<10
    GOTO 32
203:REM EFFACEMENT
     CASES MARQUEE
    S
205:FOR F=1TO 4*I
    STEP 4:X=VAL (
    MID$ (V$(0),F,
    2)):Y=VAL (
    MID$ (V$(0),F+
    2,2)):C$(X,Y)=
    ""
210:S$(X,Y)="";
    NEXT F:GOTO 20
```



Un canon (X-07) face à un mur de briques

Depuis quelque temps, le Canon X-07, l'Epson HX-20, le PB-700 et d'autres ordinateurs glissent les jeux d'arcades dans votre cartable ou votre sac à main... Parmi ces jeux, le « casse-briques » est devenu un grand classique.

■ Le programme que je vous propose pour X-07, s'il n'est pas d'une grande originalité, utilise intensivement les qualités de l'écran haute résolution. La programmation en est soignée (du moins je l'espère), depuis l'animation de texte jusqu'aux balles destructrices, en passant par un compteur et différents niveaux de difficulté.

Le jeu se déroule en trois temps : la première partie (« de plaisir » ?) consiste à taper le programme. On donne ensuite l'ordre RUN en espérant ne pas avoir introduit de bogues, ces erreurs insidieuses qui narguent tous les programmeurs. Le générique du jeu s'affiche alors.

Pour aller plus avant, frappez « I » et les règles du jeu apparaîtront à l'écran. On les fait défiler en pianotant au hasard. Mais peut-être vous sont-elles déjà familières ? Dans ce cas, après le générique, tapez une lettre autre que « I », ou demandez dès le départ RUN 1000 : le jeu commence.

Si vous ne voyez pas la balle, de deux choses l'une : vous devez soit adapter le contraste à la lumière ambiante en inclinant convenablement votre ordinateur, soit vous munir d'une bonne loupe (vos lunettes sont-elles à votre vue ?).

La liste étant claire, vous ne devriez pas avoir de difficulté pour en analyser les différentes étapes. Si le programme est long, seul ce qui suit la ligne 1000 est utile au jeu proprement dit.

Comme vous le verrez, les parties se déroulent sur un rythme assez rapide, et il est possible qu'au début, le jeu vous donne un peu de fil à retordre. Mais l'exercice aidant, vous deviendrez vous aussi un champion dans l'art de casser des briques.

☐ Jean-Marc Pluéger

Face à un mur de briques
Programme pour X-07
Auteur Jean-Marc Pluéger
Copyright *l'Ordinateur de poche* et l'auteur

```
10 CLS
20 ONERROR GOTO 410
30 DEFINTA-Z
36 'REDEFINITION DES CARACTERES
40 FONT$(130)="0,0,0,240,240,220,220,204
"
50 FONT$(131)="0,0,0,240,240,176,176,48"
60 FONT$(132)="204,192,192,192,192,192,1
92,0"
70 FONT$(133)="48,48,48,48,48,48,48,0"
80 FONT$(134)="0,0,0,192,192,192,192,192
"
100 FONT$(136)="192,192,192,192,192,252,
252,0"
110 FONT$(137)="48,48,48,48,48,240,240,0
"
120 FONT$(138)="0,0,0,252,252,196,196,25
2"
130 FONT$(139)="0,0,0,128,128,128,128,12
8"
140 FONT$(140)="252,216,216,204,204,196,
196,0"
150 FONT$(141)="128,0,0,0,0,128,128,0"
156 'M$ POUR LE GRAND M, U$ POUR LE U, R
$ POUR LE R
160 M$=CHR$(130)+CHR$(131)+CHR$(29)+CHR$
(29)+CHR$(31)+CHR$(132)+CHR$(133)
170 U$=CHR$(134)+CHR$(135)+CHR$(29)+CHR$
(29)+CHR$(31)+CHR$(136)+CHR$(137)
180 R$=CHR$(138)+CHR$(139)+CHR$(29)+CHR$
(29)+CHR$(31)+CHR$(140)+CHR$(141)
186 'DEPLACEMENT DU M DE DROITE A GAUCHE
```

```
190 FOR I=10 TO0 STEP-1
195 A$=INKEY$
196 IFA$<>"" THEN 500
200 LOCATEI,1
210 PRINTM$
220 FOR J=0TO250:NEXTJ
230 CLS
240 NEXTI
250 LOCATE 0,1:PRINTM$
260 LOCATE 2,1
270 PRINTU$
280 LOCATE 4,1
290 PRINTR$
300 LOCATE 6,2
310 PRINT"DE BRIQUE."
316 'DEPLACEMENT DE LA BALLE POUR LA PRE
SENTATION
320 X=INT(RND(X)*120):Y=0
330 DX=-1:DY=1
340 IF POINT(X+DX,Y+DY) THEN GOSUB420
350 IF X+DX<0 OR X+DX>119 THEN DX=-DX
360 IF Y+DY<0 OR Y+DY>31 THEN DY=-DY
370 PRESET (X,Y)
380 X=X+DX:Y=Y+DY
390 PSET(X,Y)
395 A$=INKEY$
396 IF A$<>"" THEN500
400 GOTO 340
406 'TRAITEMENT DE L'ERREUR SI LA BALLE
SORT
410 RESUME NEXT
416 'SOUS-PROGRAMME QUAND LA BALLE TOUCH
E UN CARACTERE
420 U=INT(RND(U)*2))
430 IFU=0THEN DX=-DX
440 IFU=1 THEN DY=-DY
450 RETURN
496 'FIN DE LA PRESENTATION
500 FOR Y=31 TO 0 STEP -1
510 LINE(0,Y)-(119,Y)
520 BEEPY,1
530 NEXT Y
540 CLS
550 IFA$<>"I"THEN CLEAR:GOTO1000
556 'LES INSTRUCTIONS
560 PRINT"*********************";
570 PRINT"   INSTRUCTIONS"
580 PRINT"*********************";
590 IFINKEY$="" THEN 590
600 PRINT"MUR DE BRIQUE:"
610 PRINT"VOUS DEVEZ DETRUIRE"
620 PRINT"UN MUR, EN DIRIGEANT";
630 PRINT"VOTRE RAQUETTE A L'";
635 IF INKEY$="" THEN 635
640 PRINT:PRINT"AIDE DES FLECHES"
650 PRINT"GAUCHE ET DROITE."
660 PRINT"POUR PASSER AU"
670 PRINT"TABLEAU SUIVANT,";
675 IFINKEY$=""THEN675
680 PRINT:PRINT"VOUS DEVEZ TOTALISER";
690 PRINT"UN SCORE DE 350 PTS.";
700 PRINT"MINIMUM."
710 PRINT"BONNE CHANCE !!!!!!";
715 IFINKEY$=""THEN715
720 CLEAR
900 '*************************************
*************************************
910 'LE PROGRAMME PRINCIPAL
998 'INITIALISATION
1000 DEFINT A-Z:GOTO 3700
1050 CLS
1100 ON ERRORGOTO 3400
1150 ZX=1:ZY=1
1200 CLS
1250 LINE(79,0)-(111,0)
1300 LOCATE14,2:PRINTA$;
1350 CONSOLE,,,0,0
1400 JMP
1450 LINE(79,0)-(79,31):LINE(78,0)-(78,3
1)
1500 LINE(111,0)-(111,31):LINE(112,0)-(1
12;31)
1546 'DESSIN DES "BRIQUES"
1550 FORIY=4TO9
1600 FORIX=81 TO 110
1650 IF (IX\5)=(IX/5)THEN 1750
1700 PSET(IX,IY)
1750 NEXT IX
1800 NEXT IY
1846 'DESSIN DE LA RAQUETTE
1850 P=94:PSET(P,29):PSET(P+1,29):PSET(P
+2,29)
1866 'COEUR DU PROGRAMME
1900 X1=ZX:Y1=ZY
1950 PSET(X,Y)
2000 GOSUB3100
2050 PRESET(X,Y)
2100 X=X+X1:Y=Y+Y1
2150 IFPOINT(X,Y)THENGOTO2250
2200 GOTO1950
2250 IF X=79ANDY=0THENX=80
2300 IF X=111 ANDY=0THEN X=110
2350 IF X=79ANDY=29 THEN X=80:Y=29
2400 IF X=111ANDY=29 THEN X=110:Y=29
2450 IFY<=0THENBEEP100,1:Y1=-Y1:GOTO2100
2500 IFY>=29 AND X=P THEN X1=-1:Y1=-1:BE
EP40,1:GOTO2100
2550 IFY>=29 AND X=P+1 THEN X1=-X1:Y1=-1
:BEEP40,1:GOTO2100
2600 IF Y>=29 AND X=P+2 THEN X1=1:Y1=-1:
BEEP40,1:GOTO 2100
2650 IF X<=79 THEN X1=-X1:BEEP100,1:GOTO
2100
2700 IF X>=111 THEN X1=-X1:BEEP100,1:GOT
O2100
2750 PRESET(X,Y):Y1=-Y1:BEEP500,1:NB=NB+
1
2800 IF NB>=350 THEN GOTO 3100
2850 LOCATE0,0:PRINTNB;
2900 IFX>80ANDINT(X)<110THENPRESET(X,Y):
PRESET(X+1,Y):GOTO3350
2950 GOTO2100
3000 PRESET(X,Y-ZY)
3050 GOTO2100
3100 B=STICK(0)
3150 IF B=0 THEN RETURN
3200 IFB=7ANDP<>80THENP=P-2:PRESET(P+3,2
9):PRESET(P+4,29)
3250 IFB=3ANDP<>108THENP=P+2:PRESET(P-1,
29):PRESET(P-2,29)
3300 PSET(P,29):PSET(P+1,29):PSET(P+2,29
):RETURN
3346 'INCREMENTATION DU SCORE
3350 NB=NB+2:GOTO1950
3396 'BALLE PERDUE
3400 SP=SP+1
3450 IFSP>3 THEN GOTO 3850
3500 LOCATE0,3
3550 PRINT"No balle:";SP;
3600 X=RND(0)*30+80:Y=10
3650 RESUME NEXT
3696 'CHANGEMENT DE TABLEAU
3700 TA=TA+1:A$="MUR"+RIGHT$(STR$(TA),1)
3750 IF NB>0 THEN SP=SP-1:NB=0
3800 GOTO1050
3850 CLS
3900 PRINT"Votre score:":PRINTNB;" ";A$
3950 END
```

Qu'y a-t-il dans une CE-150 quand on l'ouvre ?

Lorsque la CE-150 est apparue, les imprimantes des ordinateurs de poche écrivaient en une couleur sur du papier blanc ou aluminisé. Cette petite table traçante capable d'écrire et de dessiner en quatre couleurs était une sorte de bombe. Depuis, le même principe a été repris sur de nombreuses autres imprimantes pour ordinateurs de poche ou de table.

■ L'ouverture de la CE-150, l'imprimante/interface-cassettes du PC-1500, ne demande que quelques secondes. Quatre vis, au dos de la machine, solidarisent les deux corps du boîtier de plastique. Une fois ceux-ci séparés, on découvre, sur le fond, la partie mécanique de l'imprimante ; l'électronique et l'alimentation sont placées de l'autre côté. Un grand circuit imprimé en bakélite regroupe tous les composants. Dans le lot, deux circuits à haute intégration seulement. Le reste est constitué de portes logiques, de transistors, de diodes et de nombreuses résistances. Contrairement à ce que nous avions remarqué sur le PC-1500 (1), ces composants sont de conception et de dimensions classiques. Il est vrai que leur nombre est limité et que la place ne manque pas sur le circuit.

Une nappe de conducteurs souples, sur la gauche de la plaque, assure la liaison avec le PC-1500. Elle dessert également les interrupteurs de recopie de calculs, de mise hors service de la télécommande (magnétophone) ainsi que le poussoir d'avance papier.

En fonctionnement autonome, l'alimentation électrique est fournie par des batteries au cadmium-nickel. Cinq éléments de 1,2 volt sont montés en série et délivrent donc une tension de 6 volts. Des diodes réalisent une protection contre une éventuelle inversion de la tension d'alimentation. La consommation de courant est de 5,2 watts et l'autonomie d'écriture est environ de 1 100 caractères de taille standard (18 caractères par ligne).

Nous avons vu qu'il n'y avait que deux circuits à haute intégration. Ils commandent principalement le fonctionnement de la mécanique d'impression, mais ils jouent aussi un rôle dans les opérations de lecture et d'écriture sur cassettes. Le plus gros de ces deux circuits comporte 60 pattes. Il se nomme LH 5810 (ou 5811). Nous avions déjà rencontré son frère jumeau lors du démontage du PC-1500 où il gère les entrées et les sorties pour le clavier et l'écran à cristaux liquides. Même genre de tâche pour celui de la CE-150. Il reçoit les signaux provenant de l'unité centrale et les transforme en impulsions capables de commander les moteurs du système d'impression.

Une ballade en bus

Les informations sont véhiculées par le bus de données et sont interprétées grâce à un circuit de mémoire morte (le deuxième circuit multi-pattes) contenant 8 Ko situés dans la carte-mémoire à partir de l'adresse hexadécimale A000. Les codes ASCII des caractères provenant de l'ordinateur ou les intruc-

(1) Voir l'Op n° 18, page 38 : « Qu'y a-t-il dans un PC-1500 quand on l'ouvre ? ».

tions graphiques sont convertis en une représentation de séquences de mouvements qui seront transmises aux moteurs.

Dix lignes de sortie du LH 5810 effectuent ces commandes. Deux actionnent un petit solénoïde qui appuie les pointes d'écriture contre le papier. Les autres dirigent les quatre phases des deux moteurs pas à pas. Des interfaces de puissance convertissent les signaux et apportent l'intensité nécessaire au fonctionnement. Cette opération est réalisée, pour les moteurs pas à pas, par le circuit LB 1257 placé sous le connecteur en nappe. Il contient huit inverseurs.

L'interface du solénoïde comprend deux amplificateurs opérationnels contenus dans un circuit 50H001 et sept transistors assortis de résistances et de diodes de protection. Cette complexité est due au mode de commande. Le circuit LH 5810 envoie en effet des signaux de très courte durée (environ 2 millisecondes) qu'il faut convertir en tensions stables.

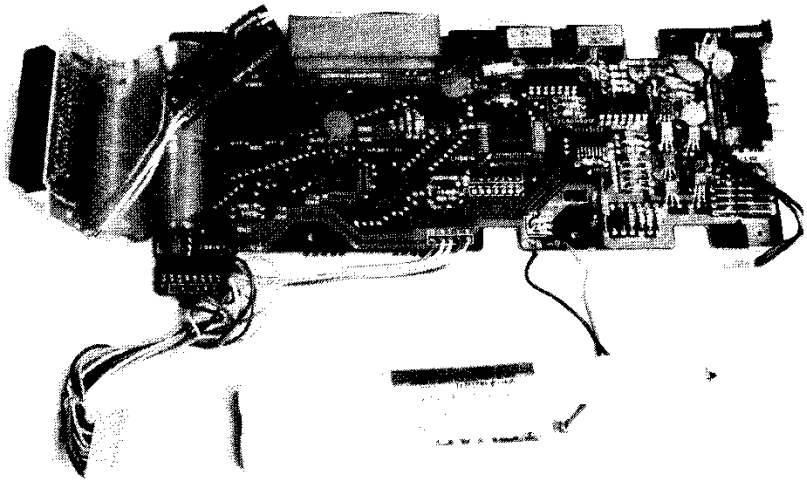
Avant d'étudier le reste de l'électronique qui compose l'interface-cassettes, regardons un instant la mécanique du système d'écriture. Le module se nomme DPG 1301. Il comprend, outre le solénoïde et les deux moteurs que nous avons déjà signalés, un barillet comportant quatre stylos, un dispositif de détection de position du barillet, des engrenages, des poulies d'entraînement et un rouleau d'avance papier.

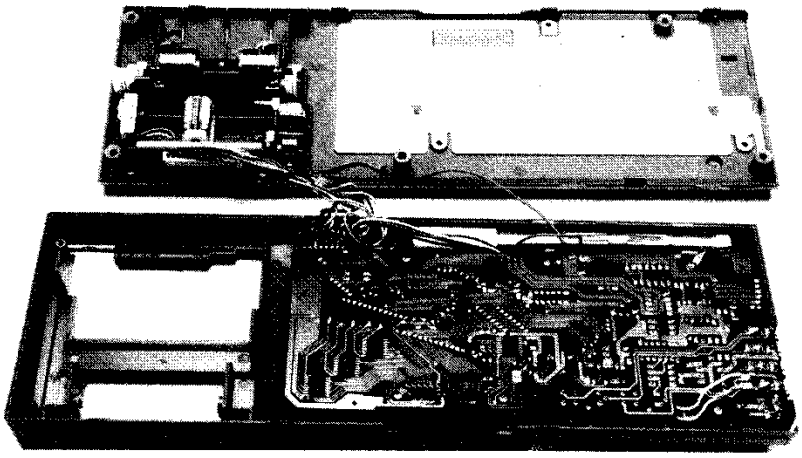
L'écriture ou le dessin sont réalisés par des mouvements conjugués du papier et d'une des pointes de stylo. Le papier est déplacé verticalement (axe Y) tandis que la tête d'écriture est guidée horizontalement (axe X). La combinaison de ces deux mouvements permet d'effectuer un tracé quelconque. Le plus petit déplacement possible est de 0,2 mm sur les deux axes.

Les deux moteurs pas à pas sont placés sous un capot. Celui de droite fait avancer le papier par l'intermédiaire d'un rouleau de caoutchouc bordé de part et d'autre par deux couronnes métalliques hérissées d'aiguilles. Les petits trous qu'elles font sont d'ailleurs visibles sur la feuille sortant de l'imprimante. Ce dispositif est complété par des presseurs à ressorts.

L'autre moteur pas à pas provoque le déplacement latéral du barillet d'écriture et sa rotation, pour le changement de couleur. Le mouvement est transmis par un jeu de poulies qui reçoivent un petit câble en acier ou en Kelvar sur les CE-150 plus récentes (le Kelvar est une fibre synthétique présentant une excellente stabilité à l'étirement, elle est utilisée, entre autres, pour confectionner les gilets pare-balles !). Toute cette timonerie se comporte comme le système déplaçant l'aiguille des stations sur les postes de radio. Pour le changement de couleur, le barillet est attiré à l'extrême gauche de sa tringle de guidage. Là, une lamelle métallique appuie sur une roue à crochet moulée dans le plastique du barillet.

Une électronique classique, exception faite de deux circuits à haute intégration.





Cinq vis à ôter et l'on sépare les deux parties du boîtier.

Du rouge au noir

Chaque fois que la tête est tirée vers la gauche, le barillet effectue 1/8e de tour. Comme l'ordinateur a pu être éteint avec une couleur quelconque en position d'écriture, il faut un système détectant la position 0 de la tête pour replacer, à la mise en route de l'ordinateur, la pointe noire en haut. Ce système de détection est constitué par un aimant, situé sur le barillet entre les pointes rouge et noire. Un contact ILS détecte le passage de l'aimant pendant la rota-

Qu'y a-t-il dans une CE-150 quand on l'ouvre ?

LIST
vous donne rendez-vous
fin juin
chez votre marchand de journaux

tion initiale et en informe le circuit LH 5810.

La CE-150 sert aussi d'interface pour les signaux de lecture-écriture sur cassette qui sont produits par le LH 5810 du PC-1500. La CE-150 ne réalise qu'une mise en forme de ces signaux. Pour l'écriture, ils sont filtrés avant de se retrouver sur la prise micro. Quatre résistances et trois condensateurs constituent un filtre passe-bande et un atténuateur qui adaptent les ondes carrées sortant du PC-1500. La différenciation des 0 et des 1 est faite en modulation de fréquence. Un 0 est produit par 4 pulsations de 1 270 Hz sur 3,5 millisecondes, tandis qu'un 1 est donné par 8 pulsations de 2 540 Hz pendant le même temps. Chaque octet est transmis en deux morceaux de 4 bits séparés par deux bits d'arrêt à 1. Un 0 est envoyé comme bit de départ et deux 1 comme signal de fin d'octet.

En lecture, la mise en forme consiste à convertir en signaux carrés les impulsions lues sur le magnétophone.

La télécommande du (ou des) magnétophone est à la charge du LH 5810 dont quatre broches aboutissent sur autant de portes inverseuses placées dans un M54516P. Elles reçoivent les signaux sous forme d'impulsions brèves (2 ms) et provoquent en sortie un état stable qui maintient le relais alimenté. Une impulsion sur une ligne le fait coller, une autre impulsion sur la seconde ligne le relâche. L'interrupteur *remote* placé sur la platine du CE-150 est branché en parallèle sur les contacts du premier relais. Il permet de supprimer l'effet de la télécommande.

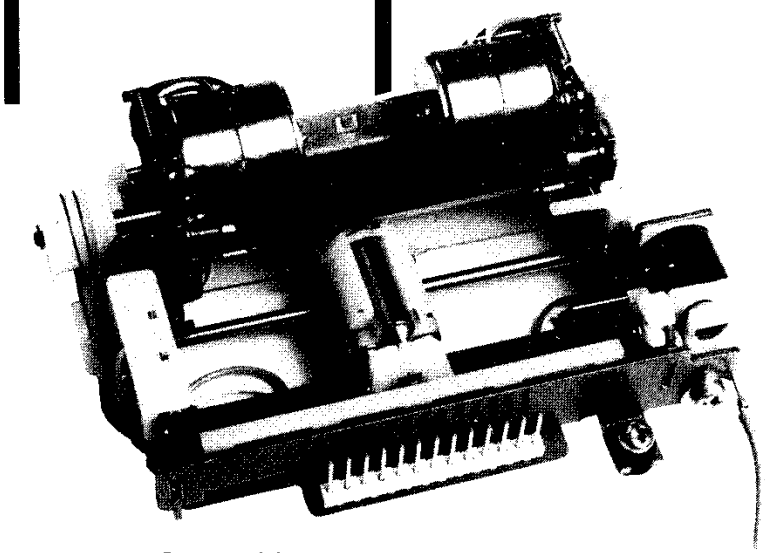
Accès aux périphériques

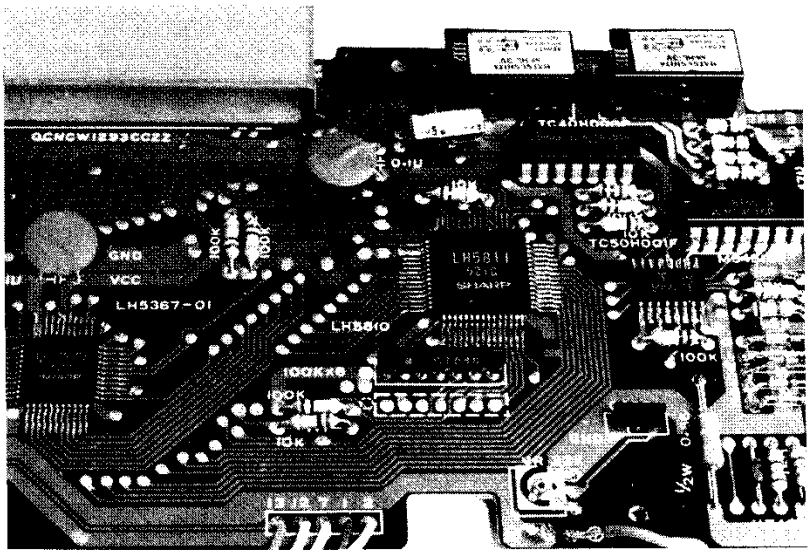
Dernière utilisation des circuits de la CE-150 : un amplificateur de bus dessert la prise placée en haut du circuit imprimé et sur laquelle on connecte éventuellement d'autres périphériques. Il est constitué de TC 50H001 visibles sur l'envers de la plaque imprimée.

La CE-150 présente des caractéristiques remarquables. On retiendra que la conception en est assez simple. La partie mécanique est limitée à des éléments présentant une bonne fiabilité et ne nécessitant pas d'entretien.

☐ Xavier de La Tullaye

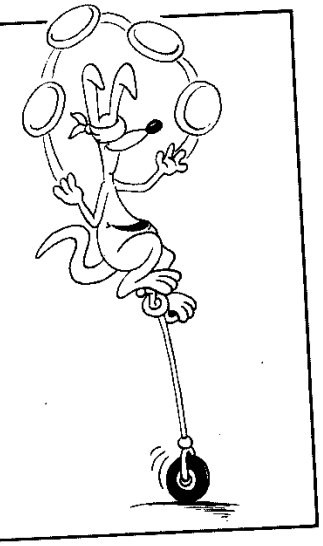
La partie mécanique de l'imprimante.





En haut de la photo, les deux relais de télécommande des magnétophones extérieurs.

Misez p'tit : Op'timisez !



Si jongler avec la pile opérationnelle de votre HP-41 C, traquer la milliseconde perdue et rogner le moindre octet est votre pain quotidien... Ou si, à l'inverse, vous échappe parfois un peu de la subtile recherche des programmes en Notation Polonaise Inverse... Alors, voici qui doit vous intéresser !

Toujours plus !

En matière de programmation, est-on jamais certain d'avoir fait aussi bien que possible ? Le mieux pourrait-il être l'ami du bien ? Dans cette rubrique, les défis des lecteurs se succèdent : des programmes toujours plus courts, encore plus rapides...

■ Arrivé à cette page de *l'Op,* sauf si vous êtes de ceux qui ne lisent *que* « Misez p'tit », vous connaissez la nouvelle : l'équipe qui fait votre journal vous donne rendez-vous fin juin, avec un nouveau titre, *LIST.* Mais votre rubrique favorite ne s'arrête pas pour autant, vous la retrouverez dans *LIST.*

Cette fois-ci, nous ne lancerons pas le défi habituel, mais donnerons en revanche la solution des défis des nos 21 et 22.

Conclusion : vous avez 60 jours pour nous concocter les défis qui rendront nombreuses les nuits blanches des lecteurs de *l'Op,* pardon, de *LIST.*

Le défi de *l'Op* n° 21, résoudre une équation du second degré, racines réelles ou complexes et discriminant « vrai » (b^2 - 4 ac), a suscité une petite avalanche de courrier ! L'équation du second degré possède à n'en point douter un charme puissant qui réveille l'instinct du coupeur-de-cheveux-en-quatre...

Et un nouveau record est établi, rappelons le texte du défi : *« résoudre une équation du second degré en rangeant la valeur du discriminant en Z et les deux solutions réelles en X et Y. Dans le cas où $\Delta < 0$, les solutions ne sont pas réelles mais complexes : u ± iv ; alors u est dans X et v dans Y ».*

Le programme de Pierre Langlois n'utilisait que les registres X, Y, Z et T de la pile opérationnelle en 34 pas et 43 octets (sans compter ni le label de tête ni le END final). A titre d'exemple, la solution de $x^2 + 1 = 0$ est donnée en 67,3 centièmes de seconde, contre 79 centièmes à la solution de $-3x^2 + 2y + 5 = 0$.

Obéir à toutes les conditions

Et malgré cette abondance de détails, une inconnue subsistait : comment fallait-il introduire les coefficients a, b et c dans la pile ? Dans l'ordre « naturel » bien sûr ! D'abord a, puis b, puis c. Cependant, tous les programmes reçus ont été considérés. En général, en dehors de a -b - c, l'ordre proposé est surtout c -b - a. Il a l'avantage de faire gagner 1 pas (2 octets) et cela apparaît nettement dans le programme vainqueur.

D'autres routines reçues au journal étaient plus courtes et plus rapides, mais ne respectaient pas une condition du problème : en Z devait se trouver le discriminant vrai $b^2 - 4ac$, dommage...

Le programme des vainqueurs, Gilles Bransbourg et Thierry Mouton, est reproduit page suivante avec le label B+M. Dans cette version, il occupe 23 lignes pour 32 octets seulement (gain de 11 octets). L'introduction des données a, b et c s'effectue bien dans l'ordre « logique » (a ENTER b ENTER c) et les solutions réelles x_1 et x_2 sont en X et Y (si elles existent), le discriminant $b^2 - 4ac$ est en Z. Dans le cas complexe, u est en X et v en Y.

La résolution de l'équation $-3x^2 + 2y + 5 = 0$ (faire −3 ENTER 2 ENTER 5 XEQ^T B + M) ne demande que 67/100e de seconde et 64/1000e (lire $x_1 = -1$, $x_2 = 1{,}66...$ et $b^2 - 4ac = 64$).

Celle de $x^2 + 1 = 0$, aux racines complexes, s'effectue en 52/100e et 29/1 000e de seconde (1 ENTER 0 ENTER 1 XEQ^T B + M, lire u = 0 et v = 1 ; le discriminant valant −4).

Petite astuce complémentaire, si les solutions sont réelles le drapeau 01 est désarmé, si elles sont complexes (forme u ± iv) il est armé.

La routine peut être modifiée pour gagner 1 pas (2 octets) en supprimant l'instruction X<>Z de départ. Les coefficients devront alors être introduits dans l'ordre c, b et a.

Enfin, supprimer le pas 14 (CF01), transformer le pas 16 en CF22 et le

Misez p'tit : Op'timisez ! HP-41

Résolution réelle et complexe d'une équation du second degré
Programme pour HP-41 C
Auteurs Gilles Bransbourg et Thierry Mouton
Copyright *l'Ordinateur de poche* et les auteurs

```
01♦LBL "B+M
"
02 X<> Z
03 CHS
04 ST/ Z
05 ST+ X
06 ENTER↑
07 X↑2
08 RDN
09 /
10 STO T
11 X↑2
12 +
13 ST* Y
14 CF 01
15 X<0?
16 SF 01
17 ABS
18 SQRT
19 ST- Z
20 R↑
21 FS? 01
22 RTN
23 +
24 R↑
25 .END.
```

pas 21 en FC?22 permet de gagner encore deux octets et 1 pas (le flag 22 est armé automatiquement lors de l'introduction depuis le clavier des coefficients).

Premier ou non ?

Le défi du mois dernier (nombres premiers) entame depuis peu sa carrière dans les kiosques à journaux au moment où ces lignes doivent être écrites. Aussi, la solution donnée n'est-elle que la meilleure *déjà* reçue ; d'autres nous parviendront, mais trop tard pour être publiées ici.

Cependant, sachant qui est l'auteur du programme vainqueur, on suppose qu'il sera difficile de faire mieux : il s'agit (encore lui !) de Gilles Bransbourg, spécialiste, s'il en est, dans l'art d'optimiser.

Son programme, reproduit ci-contre avec le label BRA, n'occupe que 16 octets ou 15 pas (sans label ni END), soit un gain d'un petit octet seulement, mais la véritable optimisation se mesure au niveau du temps d'exécution.

Ses performances sont, en effet, remarquables : le programme déclare que 65537 est un nombre premier en 28 secondes 92/100e et 21/1 000e, et que 341 est composé en 1 seconde 24/100e et 51/1 000e. Ces deux nombres servaient au test de rapidité ; le gain est ici, respectivement, de 10 secondes environ et de plus de 14 secondes : spectaculaire !

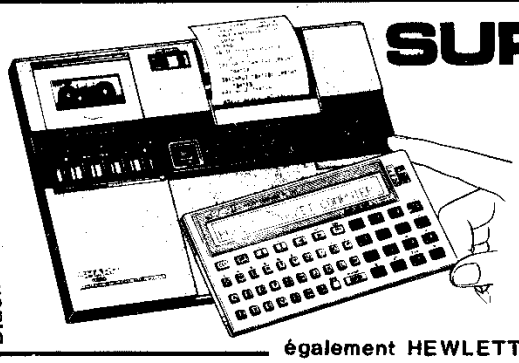
Premier ou non
Programme pour HP-41 C
Auteur Gilles Bransbourg
Copyright *l'Ordinateur de poche* et l'auteur

```
01♦LBL "BRA
"
02 ENTER↑
03 SQRT
04 1
05♦LBL 01
06 2
07 +
08 R↑
09 X<>Y
10 MOD
11 X≠0?
12 CLX
13 LASTX
14 X<Y?
15 GTO 01
16 X<>Y
17 END
```

Pour ce défi, la bataille fut encore plus rude qu'à l'ordinaire, mais l'essentiel est encore de participer ! Nous nous retrouverons dans *LIST* pour essayer encore de dépasser les limites : toujours plus, plus vite, plus court, plus beau !

☐ Jean-Christophe Krust

ORDINATEURS DE POCHE SHARP
et accessoires



SUPER PROMOTION
SUR STOCK !!
PC 1245 . PC 1255
PC 1401. PC 1251 . PC 1500
MAUBERT ELECTRONIC
49, bd. St Germain. PARIS 5e TEL. 325.88.80
également HEWLETT PACKARD-CANON-CASIO etc...
Black

Quand un poquette vous parle en morse

Il suffit de quelques minutes pour transformer le PC-1251 en un expert du morse. Dès que c'est chose faite, vous pouvez suivre son exemple et vous perfectionner dans la pratique de ce code.

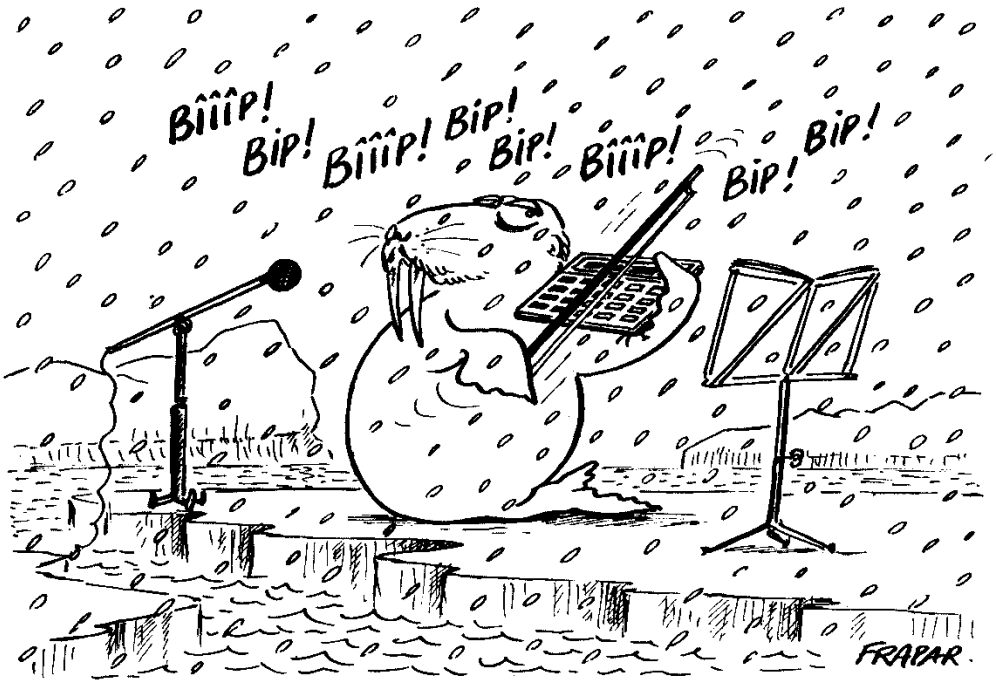
■ Nous avions déjà publié des programmes de morse pour la TI-57 et le ZX 81 (l'Op n° 20, page 25). Le codage était effectué sous forme visuelle et n'intéressait que les 26 lettres de l'alphabet.

Comme on le sait, le PC-1251, lui, n'est pas muet. Non seulement il est doté d'un bip, mais on peut aussi commander son petit haut-parleur à l'aide de l'instruction CALL. C'est donc sous forme sonore qu'il transcrira les messages.

Par ailleurs, le programme accepte l'alphabet, les chiffres de 0 à 9 et cinq signaux particuliers très utiles quand on utilise réellement le morse :

- le message d'erreur morse,
- la séquence d'appel,
- le point (fin de phrase),
- la séquence de fin de message,
- le blanc (silence pendant un court instant).

Le PC-1251 existe sous plusieurs versions qui diffèrent légèrement. Les adresses des CALL, en particulier, ne sont pas toutes identiques selon que la machine est récente ou non. Avant d'entrer au clavier le



Quand un poquette vous parle en morse

programme ci-dessous, et surtout de le faire « tourner », on devra se livrer à une petite vérification. En ligne 500, par exemple, on écrira : 500 FOR I = 1 TO 15 : CALL & 7040 : NEXT I : PRINT " " : GOTO 500

Si, à l'exécution de cette ligne, l'ordinateur se bloque, une seule solution : *All Reset*. L'adresse du CALL (& 7040) ne convient pas. Il faudra dans ce cas écrire CALL & 7071 aux lignes 90 et 91 du programme.

A noter aussi que tous les PC-1251 ne comportent pas l'inscription « ¥ », mais ils sont tous dotés de ce caractère : on l'obtient simplement en pressant les touches SHIFT et 6.

Quand on lance le programme, l'ordinateur demande quel est le message que l'on veut transcrire (79 signes maximum), puis une pression sur la touche ENTER et l'émission débute immédiatement. Maintenant que votre PC-1251 est devenu un expert en morse, il peut vous aider à le devenir à votre tour. Vous avez le choix des exercices. Au début, pour mieux identifier chaque signe du message, vous pouvez les séparer par un espace : ils seront émis nettement détachés les uns des autres par un silence.

☐ Jean-Jacques Santin

Une revue pour les amateurs de programmation :
LIST
(voir page 5)

Transcription en morse
Programme pour PC-1251
Auteur Jean-Jacques Santin
Copyright *l'Ordinateur de poche* et l'auteur

```
 5:CLEAR : DIM Q$(0)*80
   : WAIT 1
10:INPUT Q$(0): FOR J=1
   TO LEN Q$(0):A$=
   MID$ (Q$(0),J,1):
   GOSUB A$
20:FOR K=1 TO LEN C$:B$
   = MID$ (C$,K,1):
   GOSUB B$: PRINT "":
   NEXT K: PRINT "":
   NEXT J
40:GOTO 10
50:"A"C$=".-": RETURN
51:"B"C$="-...": RETURN
52:"C"C$="-.-.": RETURN
53:"D"C$="-..": RETURN
54:"E"C$=".": RETURN
55:"F"C$="..-.": RETURN
56:"G"C$="--.": RETURN
57:"H"C$="....": RETURN
58:"I"C$="..": RETURN
59:"J"C$=".---": RETURN
60:"K"C$="-.-": RETURN
61:"L"C$=".-..": RETURN
62:"M"C$="--": RETURN
63:"N"C$="-.": RETURN
64:"O"C$="---": RETURN
65:"P"C$=".--.": RETURN
66:"Q"C$="--.-": RETURN
67:"R"C$=".-.": RETURN
68:"S"C$="...": RETURN
69:"T"C$="-": RETURN
70:"U"C$="..-": RETURN
71:"V"C$="...-": RETURN
72:"W"C$=".--": RETURN
73:"X"C$="-..-": RETURN
74:"Y"C$="-.--": RETURN
75:"Z"C$="--..": RETURN
76:"1"C$=".----":
   RETURN
77:"2"C$="..---":
   RETURN
78:"3"C$="...--":
   RETURN
79:"4"C$="....-":
   RETURN
80:"5"C$=".....":
   RETURN
81:"6"C$="-....":
   RETURN
82:"7"C$="--...":
   RETURN
83:"8"C$="---..":
   RETURN
84:"9"C$="----.":
   RETURN
85:"0"C$="-----":
   RETURN
86:"*"C$=".-.-.-":
   RETURN
87:"+"C$="-.-.-":
   RETURN
88:"/"C$=".-.-.":
   RETURN
89:"£"C$=.......":RET.
90:"." CALL &7040:
   RETURN
91:"-" FOR I=1 TO 2:
   CALL &7040: NEXT I:
   RETURN
92:"¥" PRINT "": RETURN
100:" ":C$="¥": RETURN
```

Pour revoir lettre par lettre le message après sa transcription, on ajoutera la ligne suivante :

```
30 : FOR I = 1 TO LEN Q$(0) :
     PAUSE MID$ (Q$(0), I, 1) :
     NEXT I
```

Signaux particuliers

* représente un point à la fin de la phrase ;
E produit un message d'erreur en morse ;
+ produit la séquence d'appel ;
/ produit la séquence de fin de message.

Les points forts de Forth

En Forth, la programmation repose en grande partie sur un ensemble de « mots » qui s'apparentent en quelque sorte à des sous-programmes. Ces mots appartiennent au vocabulaire de base du système à moins qu'ils n'aient été créés par l'utilisateur.

■ Un ordinateur est une machine à calculer automatique à programme enregistré. Le programme définit parfaitement les différentes opérations à effectuer : à chaque code correspond une action.

Cela rappelle un peu les perforations des cartons d'orgue de barbarie. Cela dit, dans le cas de cet instrument, les ordres sont exécutés les uns à la suite des autres. On pourrait très bien imaginer une autre forme de machine à musique où la partition (enregistrée sur un support) ne serait pas toujours décodée séquentiellement. Un code supplémentaire commanderait un déplacement très rapide de la tête de lecture. Ainsi, à la fin du morceau de musique, un signal ferait remonter le lecteur au début de la bande et la machine ne s'arrêterait plus... On pourrait également commander la répétition d'une partie seulement de la partition (un refrain).

Cette belle machine théorique n'est pas une vue de l'esprit. L'électronique la rend possible. Les ordinateurs, en fait, reposent sur le même principe.

Quand on programme, on est souvent conduit à demander l'exécution de tel ou tel groupe d'instructions suivant le résultat d'un test. Si l'on trace un organigramme de l'un de ces branchements conditionnels, on obtient le dessin d'une sorte de bifurcation : chaque test conduit en somme à un aiguillage.

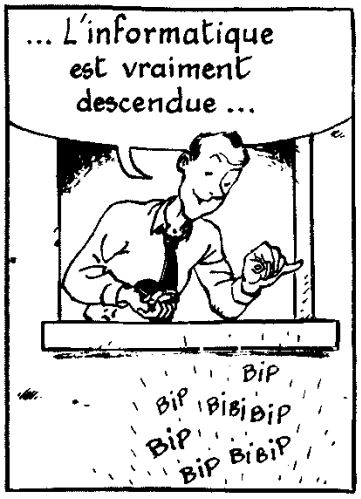
En Forth, les branchements conditionnels fonctionnent selon le schéma suivant : si la condition est vérifiée, on exécute une ou plusieurs instructions (i_1, i_2, i_3 par exemple) puis on continue par un autre groupe d'instructions j_1, j_2, j_3... Autrement dit, quand le programme rencontre IF, si la condition est remplie, il exécute toutes les instructions jusqu'à ce qu'il rencontre THEN, mais dans tous les cas les instructions situées après THEN sont exécutées : ...IF i_1 i_2... THEN j_1 j_2... Et surtout, ne pensez plus au Basic, cela ne vous aiderait pas à comprendre mais vous en empêcherait.

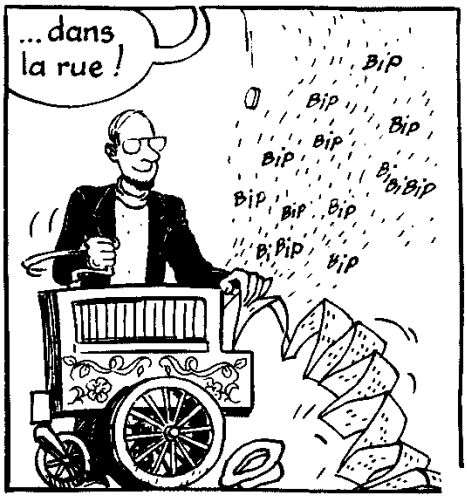
Voici, à titre d'exemple, un court programme permettant de connaître le signe d'un nombre :

```
0 : SIGNE  DUP  0< IF ." NEGATIF" ELSE DUP 0=
1                     IF ." NUL"  ELSE  ." POSITIF"
2                     THEN
3                 THEN
4  ;
```

Si le nombre est négatif, on exécute l'instruction située entre le IF et le ELSE de la ligne 0 puis les instructions comprises après le THEN de la ligne 3 qui répond au IF de la ligne 0.

Si le nombre n'est pas négatif, on exécute les instructions situées entre le ELSE de la ligne 0 et le THEN de la ligne 3. On commence donc par tester l'égalité à zéro et, s'il y a bien égalité, on exécute l'instruction





comprise entre le IF et le ELSE de la ligne 1. Dans le cas contraire, on exécute l'instruction inscrite entre le ELSE de la ligne 1 et le THEN de la ligne 2. De toutes les façons, on finit par exécuter les instructions qui suivent le THEN de la ligne 2.

En résumé, deux tests sont effectués. Le premier est construit avec IF et ELSE (ligne 0) et THEN (ligne 3). On remarque l'alignement de THEN sous l'instruction IF dont il dépend. Le second test, interne à la branche ELSE-THEN du premier, comporte IF et ELSE (ligne 1) et THEN (ligne 2). On remarque encore l'alignement de THEN.

Comme on le voit, la façon dont sont organisés les tests en Forth n'a rien à voir avec ce qu'elle est dans d'autres langages (comme le Basic par exemple). En particulier, l'utilisateur n'a pas à se préoccuper du fonctionnement de la machine, avec des sauts à commander : il suffit de suivre la logique du traitement.

Deux éléments essentiels

Si Forth est un langage élaboré, il est aussi élémentaire. Toutes les fonctions ou mots agissent suivant les mêmes règles. Partant d'un niveau très rudimentaire, on peut aboutir à un vocabulaire (ensemble de mots) très sophistiqué et adapté de façon précise aux besoins de l'utilisateur.

Par ailleurs, ce langage se prête bien à la programmation structurée. Cette manière de procéder consiste à décomposer la tâche en blocs constituant autant de segments de programme qui peuvent alors être mis au point et testés séparément. Ces blocs, à leur tour, sont décomposés : on peut donc suivre de près le cheminement logique défini par l'algorithme, ce qui limite les risques d'erreur en facilitant l'écriture et la relecture des programmes.

Les deux éléments essentiels de la machine Forth sont la pile de données et le dictionnaire (zone de mémoire où sont stockés les mots).

Quand on a bien compris le fonctionnement de la pile de données, qui contient en réalité des nombres, des adresses, des valeurs logiques, des caractères... (une vraie salade russe !), on peut regarder l'autre pile, un peu dissimulée dans le système Forth, accessible elle aussi, mais dangereuse car elle contient les paramètres des boucles et les adresses de retour des mots lors des appels successifs (une autre belle salade). Normalement, l'utilisateur ne se préoccupe pas de cette pile qui lui est parfaitement « transparente ». Mais il peut aussi, s'il la maîtrise bien, en tirer profit.

A chacun son dictionnaire

Quant au dictionnaire, il contient trois types de mots : les sous-programmes, les constantes et les variables. Dans le premier cas, les mots sont soit « primitifs » (c'est-à-dire déjà définis dans la machine), soit constitués à partir de mots primitifs. Ainsi, lorsque la machine exécute un programme, à chaque mot rencontré, elle effectue une recherche dans le dictionnaire. Si le mot est un mot primitif, il est aussitôt exécuté. Dans le cas contraire, la machine effectue une recherche analogue pour décomposer le mot et remonter jusqu'aux mots primitifs qui le définissent. Quand un mot n'a pas été défini, l'erreur est simplement signalée par le rappel du mot inconnu suivi d'un point d'interrogation. Exemple : ABCD ABCD ?

Pour accélérer la recherche dans le dictionnaire, les mots sont en fait remplacés par un code indiquant l'endroit où ils sont mémorisés. Il n'y a donc plus, à proprement parler, de recherche, mais des sauts rapides dans le dictionnaire pour remonter jusqu'aux mots primitifs. Pour chaque mot créé, le codage des positions respectives des mots qui le définissent est fait une fois pour toutes. Cette compilation s'effectue donc au fur et à mesure de l'introduction des mots ; le vocabulaire est développé suivant les besoins et si l'on modifie la définition d'un mot, il n'est pas nécessaire de tout recompiler.

Les mots entrant dans le dictionnaire sont rangés les uns à la suite des autres par ordre chronologique. Lors d'une recherche, l'exploration se fait en remontant dans la suite des mots déjà inscrits et s'arrête dès que le mot est trouvé. En conséquence, quand on redéfinit un mot, c'est toujours le dernier qui est pris en compte : les homonymes sont ignorés.

Ce qui contribue fortement à la vitesse d'exécution de Forth, à sa facilité de mise au point, c'est en définitive cette compilation effectuée morceau par morceau : chaque mot que le programmeur peut créer selon ses besoins est ramené à une codification assez proche du langage machine.

☐ Claude Nowakowski

Simplification de fractions et de racines carrées
Programme pour PC-1500
Auteur Christian Allois
Copyright *l'Ordinateur de poche* et l'auteur

```
10:REM SIMPLIFICA
   TIONS
20:IF ABS (G-INT
   G)<=1E-9*ABS G
   LET B=INT G:E=
   1:GOTO 70
30:A=1:B=1:D=0:E=
   1
40:H=1/(H-I):I=
   INT H
50:C=B*I+A:A=B:B=
   C:F=E*I+D:D=E:
   E=F
60:IF ABS (G-C/F)
   >1E-9*ABS G
   GOTO 40
70:RETURN
80:"X"AREAD R:
   WAIT :D=R*R:IF
   ABS (D-INT D)<
   =1E-9LET D=INT
   D
82:IF D=INT DLET
   B=D:GOSUB 89:
   PRINT G;"*√";H
   :END
83:G=D:H=G:I=INT
   H:GOSUB 20
84:C=√B:W=√E:IF C
   <>INT CAND W<>
   INT WGOSUB 89:
   GOSUB 95:PRINT
   G;"*√";H;"/(";
   X;"*√";Y;")";
   END
85:IF C<>INT CAND
   W=INT WGOSUB 8
   9:PRINT G;"*√"
   ;H;"/(";W;")";
   END
86:IF C=INT CAND
   W<>INT WGOSUB
   95:PRINT C;"/(
   ";X;"*√";Y;")"
   :END
87:PRINT C;"/";W:
   END
89:G=1:H=B:N=2:O=
   1
90:P=N*N:IF P>H/2
   GOTO 94
91:Q=H/P:IF INT Q
   =QLET G=G*N:H=
   Q:GOTO 91
93:N=N+O:O=2:GOTO
   90
94:RETURN
95:X=1:Y=E:N=2:O=
   1
96:P=N*N:IF P>Y/2
   GOTO 99
97:Q=Y/P:IF INT Q
   =QLET X=X*N:Y=
   Q:GOTO 97
98:N=N+O:O=2:GOTO
   96
99:RETURN
```

Racines et fractions : des résultats qui présentent bien

Extraire des racines, convertir des nombres décimaux en fractions, autant d'opérations qui permettent de simplifier et de rendre plus lisibles les résultats numériques fournis par un ordinateur. Heureusement, ce même ordinateur (ici un PC-1500 ou PC-2) peut très bien effectuer seul cette mise en forme des résultats.

■ On peut reprocher aux ordinateurs de délivrer parfois leurs résultats sous forme de valeurs numériques difficiles à interpréter. En effet, si l'on obtient, par exemple, 0,7142857143... au terme d'un calcul, on aimerait bien voir ce nombre aussitôt simplifié en 5/7 ! Mieux encore dans le cas des racines, 0,5773502692 serait avantageusement remplacé par $1/\sqrt{3}$. D'une part c'est plus joli, c'est moins long à écrire, mais c'est aussi plus satisfaisant les jours d'interrogations écrites (les professeurs n'admettent pas les réponses "à dix chiffres significatifs").

Le programme ci-contre effectue les simplifications de fractions, 256/130 devenant 128/65 et de racines, $\sqrt{45}/\sqrt{8}$ devient $3\sqrt{5}/2\sqrt{2}$. Le mode d'emploi ne présente aucune difficulté. On tape le nombre au clavier, soit sous sa forme décimale (2,371708245), soit sous une quelconque autre forme plus ou moins simplifiée ($3\sqrt{5}/\sqrt{8}$). On fait ensuite DEF X pour lancer le programme et le résultat s'inscrit sur l'afficheur : $3*\sqrt{5}/(2*\sqrt{2})$. Notez que cette expression respecte la hiérarchie des opérateurs arithmétiques du Basic.

Vous n'aurez aucune peine à adapter ce programme pour qu'il serve conjointement avec d'autres, par exemple celui d'intégration numérique des fonctions (*l'Op* n° 11 p. 48) et obtenir ainsi des résultats nettement plus présentables que de simples (?) nombres à virgule...

☐ Christian Allois

Programmation :

Jouer sur tous les tableaux

Quand on programme, les tableaux de variables offrent une solution simple à de très nombreux problèmes. C'est le cas, par exemple, pour le tri ou le mélange de données. Voici quelques algorithmes de base pour mieux comprendre le parti que l'on peut tirer des tableaux.

■ L'emploi des tableaux permet très souvent de simplifier le traitement des données. Dès que ces dernières se présentent sous une forme homogène leur conférant l'aspect d'une collection, on a le plus souvent intérêt à les regrouper en tableaux. La programmation de boucles permet alors un traitement séquentiel de la collection.

Plusieurs cas peuvent se présenter. Quand toutes les données sont définies par un seul critère, elles peuvent être rangées dans un tableau à une dimension. Si au contraire plusieurs critères doivent intervenir dans les traitements, il faudra donner autant de dimensions au tableau qu'il y a de critères utiles.

Noter en une seule dimension

Ainsi, pour suivre les notes obtenues par tous les élèves d'une classe lors d'une seule interrogation, un tableau à une dimension suffit. Pour travailler sur les notes obtenues dans plusieurs matières, il est nécessaire de définir deux dimensions, l'une représentant les élèves, l'autre les matières. Pour traiter les notes de plusieurs interrogations dans chaque matière, il faut recourir à une troisième dimension. On utilise autant de boucles qu'il y a de dimensions au tableau, et ces boucles sont de deux types différents suivant que l'on connaît à l'avance ou non le nombre d'éléments du tableau : la boucle à compteur et la boucle à itération.

L'algorithme d'utilisation de la boucle à compteur est simple à écrire puisque le nombre d'éléments du tableau est connu :

1 - Initialisation
- 1-1 entrer le nombre d'éléments
- 1-2 dimensionner le tableau

2 - Corps de la boucle
- 2-1 début de boucle : de compteur 1 à compteur maximum
  - 2-1-1 le tableau : introduction, traitement ou affichage de chaque élément
- 2-2 compteur suivant

Dans le cas d'une boucle à itéra-

tion, au lieu de faire dépendre la sortie de boucle d'une valeur de compteur, on réalise un test sur le contenu du dernier élément introduit. Si par exemple, le tableau ne doit contenir que des nombres positifs, on peut décider que l'entrée d'un nombre négatif déclenche l'arrêt du remplissage du tableau.

L'algorithme prend alors la forme suivante :

1 - Initialisation
- 1-1 entrer le nombre maximum de données traitables
- 1-2 dimensionner le tableau
- 1-3 fixer la valeur de départ de l'indice du tableau

2 - Boucle
- 2-1 entrer le contenu de la case T(I) du tableau indicé
- 2-2 test : si le contenu est égal au contenu de sortie, aller en *3*
- 2-3 I = I + 1 ; aller en *2*

3 - Suite du programme
- 3-1 afficher l'indice maximum atteint

La boucle à itération est spécialement adaptée au remplissage d'un tableau dont le nombre d'éléments est inconnu. Par la suite, pour les traitements et l'affichage, on emploiera des boucles à compteur qui se termineront sur la dernière valeur de l'indice atteinte au moment du remplissage.

Une boucle dans une autre

Si le tableau compte deux dimensions, on aura recours à deux boucles imbriquées pour en explorer les rangées et les colonnes. Pour la première rangée, on parcourt toutes les colonnes puis on passe à la rangée suivante pour laquelle on explore à nouveau toutes les colonnes, et ainsi de suite jusqu'à la dernière rangée.

L'algorithme des boucles à compteur pour de tels tableaux a la forme suivante :

1 - Initialisation
- 1-1 entrer le nombre de rangées et de colonnes maximum
- 1-2 dimensionner le tableau

2 - Boucles
- 2-1 parcours des rangées : de rangée 1 à rangée maximum
  - 2-1-1 parcours des colonnes, de 1 à maximum
    - 2-1-1-1 les éléments du tableau : remplissage, traitement ou affichage de chaque élément
  - 2-1-2 colonne suivante
- 2-2 rangée suivante

L'algorithme avec boucles à itération combine les deux boucles imbriquées de l'algorithme précédent et la sortie sur une valeur prédéfinie, comme nous l'avons vu pour le tableau à une dimension :

1 - Initialisation
- 1-1 entrer le nombre de rangées et de colonnes maximum
- 1-2 dimensionner le tableau
- 1-3 Indices de rangée et de colonne à 1

2 - Boucles
- 2-1 boucle 1 : adressage des rangées
  - 2-1-1 boucle 2 : adressage des colonnes
    - 2-1-1-1 remplir la case T(R,C) du tableau, croisement de la rangée R et de la colonne C
    - 2-1-1-2 si contenu = valeur de sortie aller à *3*
  - 2-1-2 C = C + 1 ; retour à *2-1-1* (jusqu'à indice maxi)
- 2-2 R = R + 1 ; retour à *2-1*

3 - Suite du programme
- 3-1 Afficher les indices maxi de rangée et de colonne

Convertir ces algorithmes en programmes ne devrait pas poser de problème. Pour vous guider dans vos essais, vous pouvez vous reporter au petit programme d'« utilisation d'un tableau à une dimension » effectuant les manipulations d'éléments d'un tableau à une dimension. Il correspond au premier algorithme donné. Les trois manipulations ont été placées dans des sous-programmes et sont effectuées par une boucle placée dans la racine. Aux lignes 70 à 90, GOSUB renvoie à la manipulation sélectionnée dans le menu de la ligne 40 : remplissage, traitement ou affichage du tableau. Si votre ordinateur dispose d'une instruction ON GOSUB, vous pourrez remplacer ces lignes par : ON T GOSUB 200, 300, 400. Vous placerez à la ligne 310 le traitement à effectuer. Ce peut être par exemple la somme des éléments du tableau. Dans ce cas, écrire la ligne ainsi : 310 S = S + A (C) et ajouter une ligne 415 pour l'affichage de cette somme.

Utilisation d'un tableau à une dimension

```
 10 REM *** INITIALISATION ***
 20 INPUT "NOMBRE MAXI
    D'ELEMENTS :" ; M
 30 REM *** MENU ***
 40 INPUT "REMPLISSAGE -1,
    TRAITEMENT -2, AFFICHAGE
    -3" ; T
 50 REM *** BOUCLE DE MANI
    PULATION ***
 60 FOR C = 1 TO M
 70 IF T = 1 GOSUB 200
 80 IF T = 2 GOSUB 300
 90 IF T = 3 GOSUB 400
100 NEXT C
110 GOTO 30
200 REM *** REMPLISSAGE ***
210 PRINT "CONTENU DE" ; C
220 INPUT A(C) : RETURN
300 REM *** TRAITEMENT ***
310 REM * TRAITEMENT A
    EFFECTUER, ICI *
320 PRINT "TRAITEMENT NO.:";
    C
330 RETURN
400 REM *** AFFICHAGE ***
410 PRINT "A (" ; C ;") =";A (C)
420 RETURN
```

Trier le contenu d'un tableau

Il existe de nombreuses méthodes de tri qui permettent de ranger les éléments d'un tableau dans un certain ordre (croissant ou décroissant, alphabétique...). Certaines sont simples à programmer, mais lentes à

Jouer sur tous les tableaux

exploiter. Les plus rapides ont des algorithmes assez compliqués. Nous retiendrons ici le « tri à bulles » qui présente un bon compromis entre vitesse et complexité.

Pour réaliser cet algorithme, on utilise deux boucles imbriquées. Elles permettent de comparer le premier élément du tableau à tous les autres, puis le second à ceux qui restent, etc. Chaque fois qu'un élément plus petit est rencontré, il y a échange des contenus. Pour mettre en évidence le procédé, une première liste de résultats (intitulée « Evolution du tableau, étape par étape ») montre l'évolution des contenus avant et après traitement de tri et les indices des compteurs 1 et 2 (I et J) tels qu'ils progressent à l'intérieur des boucles. On voit en première ligne l'organisation initiale du tableau, et en dernière ligne, ce qu'elle est devenue après le tri.

Evolution du tableau, étape par étape

Tableau initial			5 9 1 7 3 2
Indices des éléments traités		Eléments avant traitement	Eléments après traitement
I	J		
1	2	A(1)=5 A(2)=9	A(1)=5 A(2)=9
1	3	A(1)=5 A(3)=1	A(1)=1 A(3)=5
1	4	A(1)=1 A(4)=7	A(1)=1 A(4)=7
1	5	A(1)=1 A(5)=3	A(1)=1 A(5)=3
1	6	A(1)=1 A(6)=2	A(1)=1 A(6)=2
2	3	A(2)=9 A(3)=5	A(2)=5 A(3)=9
2	4	A(2)=5 A(4)=7	A(2)=5 A(4)=7
2	5	A(2)=5 A(5)=3	A(2)=3 A(5)=5
2	6	A(2)=3 A(6)=2	A(2)=2 A(6)=3
3	4	A(3)=9 A(4)=7	A(3)=7 A(4)=9
3	5	A(3)=7 A(5)=5	A(3)=5 A(5)=7
3	6	A(3)=5 A(6)=3	A(3)=3 A(6)=5
4	5	A(4)=9 A(5)=7	A(4)=7 A(5)=9
4	6	A(4)=7 A(6)=5	A(4)=5 A(6)=7
5	6	A(5)=9 A(6)=7	A(5)=7 A(6)=9
Tableau final			1 2 3 5 7 9

Tableau initial	5	9	1	7	3	2
Etape 1	5	9	1	7	3	2
2	1	9	5	7	3	2
3	1	9	5	7	3	2
4	1	9	5	7	3	2
5	1	9	5	7	3	2
6	1	5	9	7	3	2
7	1	5	9	7	3	2
8	1	3	9	7	5	2
9	1	2	9	7	5	3
10	1	2	7	9	5	3
11	1	2	5	9	7	3
12	1	2	3	9	7	5
13	1	2	3	7	9	5
14	1	2	3	5	9	7
15	1	2	3	5	7	9
Tableau final	1	2	3	5	7	9

Propagation des éléments, étape par étape.

La deuxième liste montre la propagation des éléments au fur et à mesure du tri.

En étudiant attentivement ces deux listes, vous devriez retrouver facilement l'algorithme du programme. En fait, c'est assez simple. L'algorithme du tri à bulles se présente comme suit :

1 - Initialisation
- 1-1 entrer le nombre maximum d'éléments du tableau
- 1-2 dimensionner le tableau

2 - Remplissage du tableau dans le désordre
- 2-1 début de boucle
  - 2-1-1 entrée d'un élément
- 2-2 jusqu'à nombre maxi, retour à *2-1*

3 - Tri
- 3-1 compteur C1 de 1 à (nombre maxi − 1)
  - 3-1-1 compteur C2 de C1 + 1 à nombre maxi
    - 3-1-1-1 Si le contenu de la case T(C2) < au contenu de la case T(C1), échanger les contenus
  - 3-1-2 C2 suivant
- 3-2 C1 suivant

4 - Affichage
- 4-1 boucle de présentation identique à *2*

L'algorithme se convertit aisément en programme. Le seul point noir réside dans l'absence sur les Basics des ordinateurs de poche d'une instruction SWAP (présente parfois sur de plus gros ordinateurs) qui sert à échanger le contenu de deux variables. Pour la remplacer, il faut recourir à trois affectations successives. Une variable intermédiaire (X dans le programme « tri à bulles ») reçoit le contenu de l'élément 1. Celui-ci est ensuite rempli avec le contenu de l'élément 2 qui est alors rechargé avec la variable intermédiaire. Vous trouverez ci-dessous la liste du programme. Vous pourrez ainsi vous familiariser avec le tri à bulles. Et à partir de là, explorer et réinventer d'autres méthodes.

Mélanger le contenu d'un tableau

Les algorithmes de mélange sont moins répandus que ceux de tri. Ils présentent pourtant un intérêt non négligeable dans la programmation de nombreux jeux. C'est le cas, par exemple, quand il faut simuler le mélange d'un jeu de cartes. Si vous vous contentez de tirer des nombres aléatoires, vous avez beaucoup de chances de trouver plus de quatre

Tri à bulles

```
10 INP "NB DONNEES
   ",N
20 FOR I=1 TO N
30 INP A(I)
40 NEXT I
50 FOR I=1 TO N-1
60 FOR J=I+1 TO N
70 IF A(J)<A(I):X=
   A(I):A(I)=A(J):
   A(J)=X
80 NEXT J:NEXT I
90 FOR I=1 TO N
100 PRT A(I);
110 NEXT I
```

as dans le jeu. Un mélange effectué sur un tableau représentant le jeu de cartes simulera beaucoup mieux la réalité.

Un mélange au hasard

Le principe du mélange repose aussi sur un générateur de nombres aléatoires. On procède seulement de telle sorte qu'un nombre ne soit pas tiré deux fois. Ou plus exactement, comme on opère sur les indices d'un tableau, on s'arrange pour échanger le contenu d'un élément qui a déjà été tiré avec celui d'un élément qui ne l'a pas encore été.

Il est donc nécessaire de disposer de deux tableaux : un tableau de départ contenant les données à mélanger et un tableau d'arrivée. Voici l'algorithme du système :

1 - Initialisation
- 1-1 entrer le nombre maxi d'éléments
- 1-2 dimensionner deux tableaux (Tableau Départ et Tableau Arrivée)

2 - Remplissage du tableau de départ
- 2-1 utilisation d'une boucle compteur

3 - Mélange
- 3-1 début de boucle : Indice 1 varie du nombre maxi d'éléments à 1
  - 3-1-1 tirage de Indice 2 (nombre aléatoire) compris entre 1 et Indice 1
  - 3-1-2 Le Tableau Arrivée est rempli par l'élément d'Indice 2 du Tableau Départ
  - 3-1-3 échange dans le Tableau Départ des éléments d'Indice 1 et d'Indice 2
- 3-2 suite de la boucle, retour à *3-1* tant que Indice 1 > 1

4 - Présentation du tableau mélangé (Tableau Arrivée)
- 4-1 Boucle d'affichage identique à *2-1*

Voyons, dans le détail, comment se déroulent les opérations. Un tableau de départ TD, unidimensionnel et contenant sept éléments, est initialisé : TD 1 2 3 4 5 6 7. Un tableau d'arrivée TA, de même taille que TD et qui contiendra le « mélangé », est lui aussi initialisé, TA 0 0 0 0 0 0 0.

L'algorithme de mélange impose le passage par une boucle qui aura plusieurs effets :

- initialiser deux indices I1 et I2, I1 variant de 7 à 1 et I2 étant un nombre aléatoire compris entre 1 et I1.
- donner à la case I1 du tableau d'arrivée la valeur de la case I2 du tableau de départ
- donner à la case I2 du tableau de départ la valeur de la case I1 de ce tableau TD (afin de ne plus rencontrer la valeur tirée et mise dans TA).

Et dans le détail

A la première étape, I1 prend la valeur 7 et I2, la valeur 1 (tirée au hasard). Alors, la case 7 de TA prend la valeur de la case 1 de TD, soit 1 : TA 0000001. Puis la case 1 de TD prend la valeur de la case 7, soit 7 justement, les autres cases restant intactes : TD 7 2 3 4 5 6 7.

A la deuxième étape, I1 passe à 6 et I2, tirée au hasard entre 1 et 6 prend — disons — la valeur 1. Alors, la case 6 de TA doit prendre la valeur de la case 1 de TD, ici c'est 7 : TA 0000071. Et la case 1 de TD prend la valeur de la case 6, soit 6 : TD 6234567.

A la troisième étape, I1 = 5 et I2 prend une valeur aléatoire comprise entre 1 et 5, disons 4. Alors, la case 5 de TA prend la valeur de la case 4 de TD, soit 4 : TA 0000471. Et la case 4 de TD prend la valeur de la case 5, les autres cases restant toujours intactes : TD 6235567.

Le processus se répète ensuite :

- quatrième étape, I1 = 4, I2 = 2, d'où TA 0002471 et TD 6535567
- cinquième étape, I1 = 3, I2 = 3, d'où TA 0032471 et TD 6535567
- sixième étape, I1 = 2, I2 = 1, d'où TA 0632471 et TD 5535567

A la dernière étape, I1 = I2 = 1 (seule valeur possible) et TA 5632471 et TD 5535567. TA contient bien le tableau initial dans le désordre.

Partant de ce dernier algorithme, le programme coule de source. Vous pourrez adapter la liste qui suit à votre ordinateur de poche.

Le jeu de Loto va nous aider à mettre en évidence l'intérêt des mélanges par tirages aléatoires. Vous connaissez le principe de ce jeu. Il consiste à tirer six numéros et un numéro complémentaire dans un ensemble fini de 49 éléments. Un numéro ne peut pas être tiré deux fois. Si vous voulez programmer un tel jeu avec un tirage aléatoire classique, vous devrez après chaque tirage comparer le nombre produit avec tous les numéros déjà tirés. En cas d'égalité, le tirage sera recommencé. La structure du programme est assez compliquée. Vous pouvez aussi tirer l'ensemble des 7 numéros puis tester ensuite tous les élé-

Dorénavant, écrivez à

LIST

la revue de la programmation individuelle

5 place du Colonel Fabien
75491 PARIS CEDEX 10

```
 10 REM *** INITIALISATION ***
 20 INPUT "NOMBRE MAXI
    D'ELEMENTS"; M
 30 DIM Y(M), Z(M)
 40 REM *** REMPLISSAGE
    TABLEAU DEPART ***
 50 FOR C = 1 TO M
 60 Y (C) = C
 70 NEXT C
 80 REM *** MELANGE ***
 90 FOR D = M TO 1 STEP -1
100 E = INT (RND(0)*D) +1
110 Z (D) = Y (E)
120 Y (E) = Y (D)
130 NEXT D
140 REM *** AFFICHAGE ***
150 FOR C = 1 TO M
160 PRINT Z (C)
170 NEXT C
180 END
```

Mélange des éléments d'un tableau

Jouer sur tous les tableaux

ments. Cela ne fait pas moins de 42 tests successifs à programmer. Bien sûr, le programme tournera, mais il ne sera pas très élégant. Et si par la suite, vous voulez le modifier pour qu'il vous aide à remplir des grilles multiples, il faudra ajouter d'autres tests.

L'algorithme de mélange des éléments d'un tableau produit une structure de programme beaucoup plus simple. Si l'on veut d'ailleurs que le programme donné précédemment devienne un très bon tireur de Loto, on remplacera la ligne 20 par 20 M = 49, et à partir de la ligne 140, on écrira une nouvelle routine d'affichage :

```
140 REM *** AFFICHAGE DU TI
    RAGE ***
150 PRINT « TIRAGE DU LOTO »
160 PRINT « LES SIX BONS NUME
    ROS SONT : »
170 FOR C = 1 TO 6
180 PRINT Z (C)
190 NEXT C
200 PRINT « »
210 PRINT « LE NUMERO COMPLE
    MENTAIRE EST : » ; Z (7)
```

Pour que le programme remplisse des grilles de jeu, il suffit de modifier cette dernière routine.

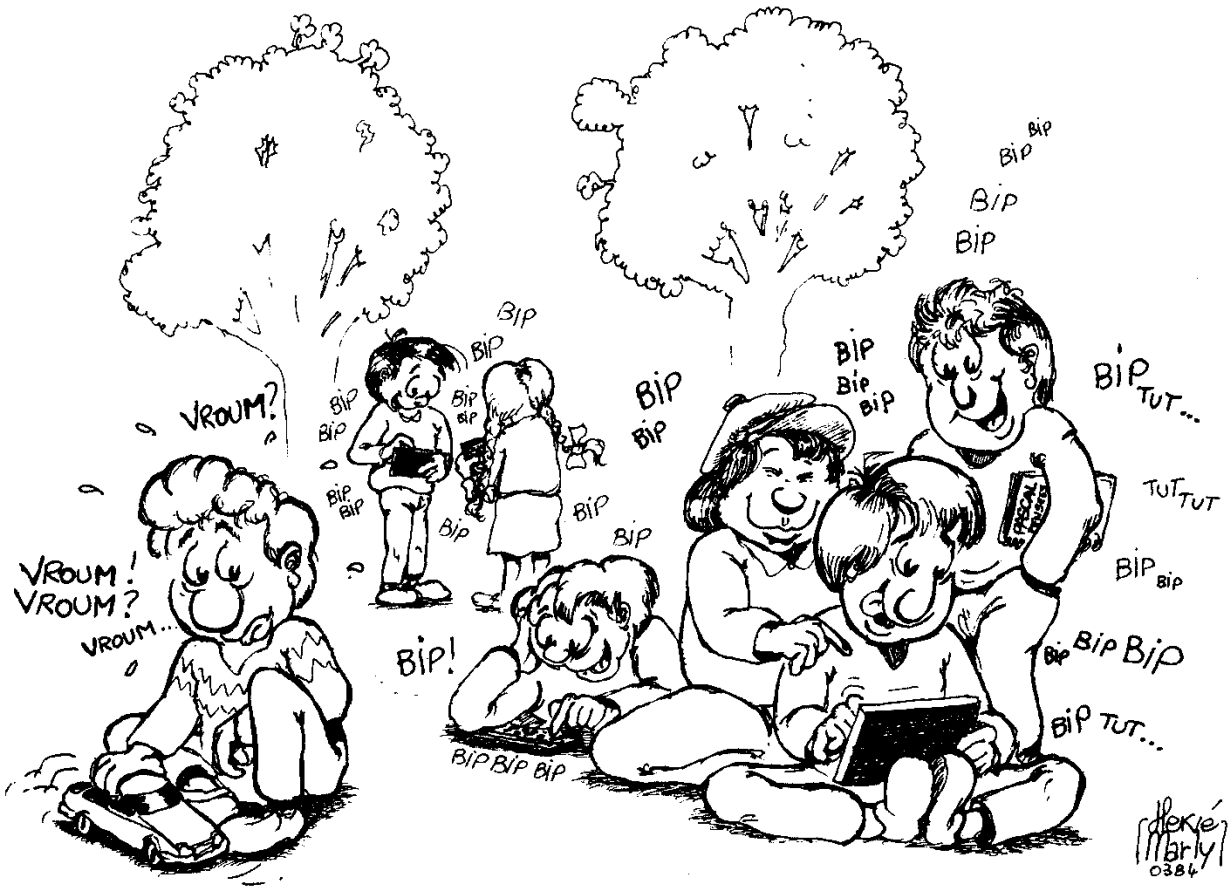
```
140 REM *** REMPLISSAGE DES
    GRILLES ***
150 INPUT « NOMBRE DE CHIF
    FRES POUR REMPLIR LA
    GRILLE » ; N
160 FOR C = 1 TO N
170 PRINT Z (C)
180 NEXT C
190 PRINT « »
200 PRINT « BONNE CHANCE ! »
```

Dans le cas où vous jouez 8 grilles simples, vous pouvez d'ailleurs ajouter une boucle qui parcourt 8 fois les lignes placées après la ligne 40.

Le même algorithme de mélange peut être utilisé pour battre un jeu de cartes. L'adaptation est un peu plus compliquée car on doit tenir compte des différentes couleurs. Dans certains jeux, on pourra avoir des cartes différentes ayant la même valeur. Il faudra utiliser des tableaux à plusieurs dimensions et éventuellement opérer le mélange sur les indices du tableau plutôt que sur le contenu.

Comme on le voit, dans ce domaine, ce ne sont pas les exercices de programmation ni les applications qui manquent.

☐ Xavier de La Tullaye



Ah ! si vous aviez su...

Vous ne connaissez pas votre machine à fond, et moins encore les autres machines... Ces quelques « ficelles » devraient vous aider.

Au bon endroit de la cassette (PC-1211)

Un magnétophone à cassettes ne possède pas toujours de compteur de tours. C'est très ennuyeux, surtout si l'on veut positionner une cassette au début d'un programme. Si le magnétophone obéit à la télécommande pour la lecture, l'enregistrement, mais aussi pour l'avance rapide, le remède est simple.

La solution se présente sous la forme d'un court programme, une sorte d'utilitaire pour le PC-1211, qui positionne la cassette sur un secteur choisi. Son chargement étant rapide, il sera bien utile de le sauver au début de chaque cassette. Et comme il est écrit sur les lignes 991 à 996, il a peu de chances de déborder sur d'autres programmes.

Quand il est chargé, on tape SHFT SPC et l'ordinateur demande le « NUMERO DE SECTEUR ? = ». Avec une cassette 90 minutes, on peut définir 22 secteurs pouvant contenir chacun un programme d'un Koctet. Il faudra modifier ces données, au besoin, en fonction du magnétophone utilisé.

Le numéro de secteur étant introduit et suivi de l'appui sur ENTER, l'ordinateur répond « AVANCE RAPIDE !!! » pour rappeler qu'il faut mettre le magnétophone sous ce mode. A nouveau ENTER et le message « MEM EN Y » apparaît. Il faut alors mettre en Y le nombre de mémoires utilisables qui est lu en tapant l'ordre MEM. C'est-à-dire qu'il faut taper MEM (ENTER), lire le nombre de mémoires et taper : Y = nombre de mémoires (ENTER). Un nouvel appui sur ENTER permet au magnétophone de se mettre en marche. A la fin de l'opération, l'ordinateur indique « SECTEUR N. x ». Il ne reste plus qu'à sauver ou charger un programme que l'on retrouvera sans difficultés. Le plus simple est encore d'appeler chaque programme par son numéro de secteur : « P1 », « P2 », etc.

Cet utilitaire de positionnement automatique est basé sur l'instruction PRINT # qui peut à la fois mettre en marche le magnétophone et charger des mémoires de données à partir de la mémoire précisée (ici A (Y) ; voir ligne 994). Quand cette mémoire A (Y) est la dernière utilisable, le chargement dure environ six secondes. Si le magnétophone est en mode « avance rapide », l'enregistrement de la variable A (Y) n'a pas lieu, et l'instruction PRINT # télécommande six secondes d'avance rapide. Facile et pratique, il sera difficile, pour certains, de s'en passer.

☐ Frank Broutin

Positionnement automatique
Programme pour PC-1211
Auteur Frank Broutin
Copyright *l'Ordinateur de poche* et l'auteur

```
991:" ":INPUT "N
    UMERO DE SEC
    TEUR?=";Z
992:BEEP 2:PRINT
    "AVANCE RAPI
    DE !!!":BEEP
    2:PRINT "MEM
     EN Y":Y=Y+2
    6
993:FOR X=1TO Z
994:PRINT #"L";A
    (Y)
995:NEXT X
996:PRINT "SECTE
    UR N.";Z
```

Un bip-bip pour les ZX 81

Voici un procédé qui permet de faire émettre un bip au ZX 81 quand on le lui demande. Il repose sur le fait que le passage du mode SLOW au mode FAST produit un petit bruit sec, un « tic », sur le haut-parleur du téléviseur. Une série de tics, et l'on obtient sans difficulté le bip recherché. En ajoutant des instructions REM, on peut d'ailleurs jouer sur la fréquence de ce bip.

Dans le court programme donné à titre d'exemple, on inverse au besoin les lignes 20 et 30 selon que l'on veut retrouver le mode FAST ou SLOW à la fin de la routine.

On pourra bien entendu faire de cette routine un sous-programme et modifier la valeur de fin de boucle pour agir sur la durée du bip.

☐ Jean-Christophe Laune

Bip-bip
Programme pour ZX 81
Auteur Jean-Christophe Laune
Copyright *l'Ordinateur de poche* et l'auteur

```
10 FOR X=0 TO 100
20 FAST
30 SLOW
40 NEXT X
```

Quatre lignes, et le ZX 81 n'est plus muet.

Les mots de passe-passe sur FX-702 P

Quand un programme pour FX-702 P a été enregistré sur cassette, une courte instruction permet de le charger dans l'ordinateur : LOAD. Et le transfert a lieu, sans problème. Mais pour lister ce programme, lorsqu'il est protégé par un mot de passe que vous avez, bien sûr, oublié et que vous souhaitez, bien sûr, retrouver, les choses se compliquent.

Deux cas peuvent se présenter, ou bien le programme avait été sauvé sur cassette par l'instruction SAVE, ou bien il l'avait été par l'instruction SAVE ALL. S'il avait été enregistré par SAVE, une méthode rapide (déjà présentée dans *l'Op* 17 page 20) permet de retrouver le mot de passe :

- faire CLR ALL en mode 1 et entrer en P0 la ligne « TOTO » (10 TOTOTO etc.) faite de T et de O jusqu'à saturation (frapper la lettre T puis la lettre O et non l'instruction TO) ;
- après avoir frappé EXE, éteindre et rallumer la machine ;
- charger alors le programme en P1, par LOAD # 1 et faire LIST ALL.

La ligne TOTO introduite en zone P0 défile. Elle a changé d'allure et se termine même par le mot de passe retrouvé.

Si le programme protégé avait été enregistré par un SAVE ALL, on doit agir un petit peu différemment pour

découvrir le mot de passe. Après le CLR ALL en mode 1, il faut charger le programme par un LOAD ALL puis sur une autre cassette (ou l'autre face de la cassette initiale), sauvegarder le programme protégé au moyen d'un simple SAVE.

Il est inutile, ici, de sauvegarder tout le programme puisque ce qui nous intéresse est le mot de passe et qu'il se trouve au début. Dix secondes d'enregistrement suffiront donc.

Cette opération terminée, il reste à refaire CLR ALL, à entrer en P0 la ligne TOTO et à éteindre puis à rallumer la machine. Alors seulement, vous pouvez charger le programme (ou morceau de programme) enregistré sur la deuxième cassette en faisant LOAD # 1. En demandant LIST ALL, la ligne TOTO transformée défile, suivie là encore du mot de passe.

On peut alors se demander si protéger un programme sur FX-702 P est encore vraiment nécessaire...

☐ Curt Knabe

Quelques "CALL" du PC-1500

En vrac, quelques adresses de routines en langage-machine pour le PC-1500 : CALL 52123 = SHIFT MODE (met en mode Réserve) ; CALL 52128 = MODE (bascule de PRO et RUN) ; CALL 52279 = touche ↑ ; CALL 52303 = touche ↓.

Enfin, les deux plus beaux : tout d'abord, CALL 52122 est comme un END en Basic mais qui conserverait un message affiché ; CALL 58188, enfin, met le PC-1500 en attente de la pression d'une touche ; l'ordinateur repart dès que la touche est enfoncée. On peut utiliser n'importe quelle touche sauf SHIFT, DEF, SML et ON.

☐ Jérôme Gaudin

Une routine pour faire bonne impression

Quand on utilise l'imprimante PC-100 de la TI-58 ou 59 pour rédiger quelques lignes de texte, le sous-programme « PRT » simplifie agréablement la tâche ; il demande et stocke dans les registres appropriés les (4 × 5) codes qui composent une ligne, et l'entrée de la dernière donnée est suivie de l'impression. Il suffit donc d'écrire le texte en continu. Pour lancer ce petit utilitaire, faire simplement SBR 2nd PRT.

☐ Philippe Tenant

Bonne impression
Utilitaire pour TI-58/59 et PC-100
Auteur Philippe Tenant
Copyright *l'Ordinateur de poche* et l'auteur

000	76	LBL	012	69	OP	024	01	01
001	99	PRT	013	03	03	025	25	CLR
002	71	SBR	014	71	SBR	026	91	R/S
003	53	(	015	53	(	027	65	×
004	69	OP	016	69	OP	028	01	1
005	01	01	017	04	04	029	00	0
006	71	SBR	018	69	OP	030	00	0
007	53	(	019	05	05	031	85	+
008	69	OP	020	76	LBL	032	91	R/S
009	02	02	021	53	(	033	95	=
010	71	SBR	022	04	4	034	97	DSZ
011	53	(	023	42	STO	035	01	01
						036	00	00
						037	27	27
						038	92	RTN

Seule la mémoire 01 est utilisée

Un pot commun pour toutes les machines

X-07 pour glisser sur la bonne pente

Ce petit programme permet d'exploiter la taille de l'écran du X-07 : on y dessine plusieurs courbes, en deux dimensions, en fonction de l'intervalle d'étude sur l'axe des X et des Y introduit par INPUT ; on peut bien sûr donner un aspect plus « courbe » au tracé en reliant entre eux les points caractéristiques de la courbe ; pour cela modifier les lignes 100, 120 :

```
100 LINE (0,0) — (0,31) : FOR I =
    0 TO 119
120 IF YP < 31 ANDYP > 0 THEN
    LINE — (XP,YP)
```

Cependant, seuls les points effectivement sur l'écran seront reliés. C'est pourquoi la courbe serait fausse dans le cas de points hors échelle. Pour faire tourner cet utilitaire, il suffit d'utiliser la fonction « DEFFND (X) = » entre les lignes 10 et 80.

Par exemple, pour étudier simultanément les fonctions (cos X), cos (X

Ligne 3	détermine l'échelle
80	dessine les axes
100	autant d'itérations que de points
110	calcule y = F(x)
115	ajuste y à l'échelle de l'écran
120	imprime y si possible

Le programme dans ses grandes lignes

+ π), cos (X + π/2) et (sin X), il suffit d'enregistrer aux lignes 10 et suivantes :

```
10 DEFFND (X) = COS (X) :
   GOSUB 100
20 DEFFND (X) = COS (X +
   3,1416) :GOSUB 100
30 DEFFND (X) = COS (X +
   1,5708) : GOSUB 100 :
   DEFFND (X) = SIN (X) :
   GOSUB 100
```

(n'oubliez pas de les effacer la fois suivante).

Puis RUN ↴
— 13 ↴ 6 ↴ — 1,2 ↴ 1,2 ↴

et l'écran se remplit rapidement d'arabesques sinusoïdes qui sont le résultat de la superposition des courbes représentatives.

☐ Xavier Benigni

Tracé de courbes
Programme pour X-07
Auteur Xavier Benigni
Copyright *l'Ordinateur de poche* et l'auteur

```
  1 INPUT "Xmin Xmax Ymin
    Ymax" ; XN, XM, YN, YM
  3 A = ABS (XM — XN) :
    B = ABS (YM — YN) : CLS
...
 80 C = ((YN/B) + 31) × 31 : IF
    (YN × YM) < 0 THEN LINE (C,
    0) — (C, 31)
 90 GOTO 90
100 FOR XP = 0 TO 119
105 X = (XP × A/119) + XN
110 Y = FND (X)
115 YP = 31 × (1 — ((Y — YN)/B))
120 IF YP < 31 AND YP > 0 THEN
    PSET (XP, YP)
125 NEXT P : RETURN
```

Sur les autres machines

Plusieurs programmes de tracé de courbes ont été publiés dans nos colonnes :

HP-41 C	*l'Op* 2 page 26
TI-58/59	*l'Op* 2 page 65
FX-702 P	*l'Op* 6 page 74
PC-1500	*l'Op* 8 page 65
PC-1211	*l'Op* 8 page 74
FX-602 P	*l'Op* 22 page 56

Kibur
pour PB-100

■ Sans extension mémoire, puisque 453 pas suffisent, le PB-100 propose un jeu de Kibur.

Sur son petit écran, dix lettres apparaissent. Elles sont en désordre. Le but du jeu est de les remettre dans l'ordre... alphabétique. La méthode qui permet d'y arriver est rappelée dans l'encadré ci-dessus.

Adapté au PB-100, ce jeu, déjà rencontré sur d'autres ordinateurs de poche (1), est souple à l'emploi. Vous pouvez choisir entre une combinaison tirée par la machine (choix 1), une combinaison que vous proposez vous-même (choix 2) ou encore rejouer avec la combinaison de la partie précédente (choix 3).

Quand les dix lettres sont dans le désordre, on désigne la position à jouer, entre 0 et 9, en pressant sur la touche correspondante. Et quand enfin l'ordre est atteint, le résultat est affiché sous la forme d'un nombre de coups.

☐ Raoul Lebastard

Kibur : le principe

Les dix premières lettres de l'alphabet, de A à J, sont proposées mélangées. Il faut les réordonner en un minimum de coups.

La méthode...

Chaque lettre occupe une position numérotée (de 0 à 9) de la gauche vers la droite. Un coup est défini par l'interversion de la lettre dont on indique la position avec celle qui occupe la position 0. L'effet secondaire de chaque coup est d'intervertir également les deux lettres situées de part et d'autre de celle qui a été désignée pour être intervertie avec la lettre en position 0.

Ainsi, jouer 5 avec la combinaison FBCDGAEHIJ, autrement dit jouer la lettre A, retourne la combinaison ABCDEFGHIJ. Les lettres F et A ont pris la place l'une de l'autre, mais aussi les lettres G et E qui entouraient la lettre A dans la combinaison initiale. Le programme considère que les positions 0 et 9 sont adjacentes, et les effets secondaires interviennent également lors du jeu de ces coups.

Kibur
Programme pour PB-100
Auteur Raoul Lebastard
Copyright *l'Ordinateur de poche* et l'auteur

```
453 Pas
10 R=0:INPUT "Choi
   x 1/2/3",B:IF B
   =3:GOSUB 55:GOT
   O 29
13 IF B=2:INPUT $:
   GOSUB 55:GOTO 2
   9
20 VAC :$="ABCDEFG
   HIJ"
22 FOR Y=0 TO 9
23 T=INT (RAND*10)
   +1:D$(Y)=MID(T,
   1)
24 FOR Z=0 TO 9:IF
   D$(Y)=D$(Z):IF
   Y≠Z THEN 23
25 NEXT Z:NEXT Y
26 $="":FOR Y=0 TO
   9:$=$+D$(Y):NE
   XT Y
29 O$="":FOR N=0 T
   O 6:O$=O$+D$(N)
   :NEXT N
30 P$="":FOR N=7 T
   O 9:P$=P$+D$(N)
   :NEXT N
35 IF O$+P$="ABCDE
   FGHIJ":PRINT :P
   RINT R;" Coups"
   :END
36 PRINT CSR 0:O$+
   P$;:A$=KEY:IF A
   $="" THEN 36
37 R=R+1:A=VAL(A$)
   :V$=D$:D$=D$(A)
   :D$(A)=V$
38 IF A=9:Q=0:B=0:
   GOTO 40
40 IF A=0:B=9:Q=1:
   GOTO 48
44 Q=A+1:B=A-1
48 V$=D$(Q):D$(Q)=
   D$(B):D$(B)=V$:
   GOTO 29
55 FOR X=1 TO 10:C
   $(X)=MID(X,1):N
   EXT X:RETURN
```

Sur les autres machines

Plusieurs programmes de tracé de courbes ont été publiés dans nos colonnes :

HP-41 C	*l'Op* 2 page 26
TI-58/59	*l'Op* 2 page 65
FX-702 P	*l'Op* 6 page 74
PC-1500	*l'Op* 8 page 65
PC-1211	*l'Op* 8 page 74
FX-602 P	*l'Op* 22 page 56

(1) Sur HP-41 dans l'Op *3 page 42, sur TI-58/59 dans* l'Op *4 page 64, sur TI-57 dans* l'Op *8 page 73, sur FX-702 P dans* l'Op *10 page 65, sur PC-1211 dans* l'Op *13 page 68, sur PC-1500 dans* l'Op *16 page 60.*

Machine à sous
pour TI-58/59

■ A ce jeu, vos gains sont symboliques, mais en contrepartie, vous n'avez rien à débourser. Malgré cela, pour faire «plus vrai», on admettra que chaque partie coûte un franc (1).

Comment fonctionne la machine ? Imaginez une fenêtre horizontale comportant trois carreaux. Derrière ces carreaux se trouvent trois roues où sont inscrits les dix chiffres de 0 à 9. Les trois roues sont lancées puis s'arrêtent au hasard, affichant un nombre compris entre 0 et 999 qui constitue la combinaison initiale.

Vous consultez alors la liste des combinaisons gagnantes et sur les trois chiffres tirés, vous devez au moins en retenir un, c'est-à-dire bloquer une roue au minimum. Puis vous relancez la machine, et la ou les roues que vous n'avez pas retenues tournent de nouveau avant de

(1) Des jeux similaires ont été proposés pour le PC-1500 dans le n° 13 de l'Op *page 60, pour le PB-100 dans le n° 16 page 63 et pour HP-41 C dans le n° 22 page 55.*

Machine à sous
Programme pour TI-58/59
Auteur Pierre Flener
Copyright *l'Ordinateur de poche* et l'auteur

000	22	INV	049	00	00
001	58	FIX	050	43	RCL
002	22	INV	051	00	00
003	97	DSZ	052	29	CP
004	05	05	053	67	EQ
005	00	00	054	95	=
006	16	16	055	69	OP
007	58	FIX	056	30	30
008	03	03	057	87	IFF
009	43	RCL	058	01	01
010	05	05	059	00	00
011	55	÷	060	64	64
012	03	3	061	18	C'
013	22	INV	062	42	STO
014	28	LOG	063	01	01
015	85	+	064	87	IFF
016	43	RCL	065	02	02
017	00	00	066	00	00
018	95	=	067	71	71
019	94	+/-	068	18	C'
020	91	R/S	069	42	STO
021	76	LBL	070	02	02
022	18	C'	071	87	IFF
023	36	PGM	072	03	03
024	15	15	073	00	00
025	71	SBR	074	78	78
026	88	DMS	075	18	C'
027	65	×	076	42	STO
028	01	1	077	03	03
029	00	0	078	43	RCL
030	95	=	079	01	01
031	59	INT	080	52	EE
032	92	RTN	081	02	2
033	76	LBL	082	85	+
034	17	B'	083	43	RCL
035	43	RCL	084	02	02
036	04	04	085	52	EE
037	66	PAU	086	01	1
038	66	PAU	087	85	+
039	66	PAU	088	43	RCL
040	66	PAU	089	03	03
041	66	PAU	090	95	=
042	92	RTN	091	22	INV
043	76	LBL	092	52	EE
044	14	D	093	22	INV
045	87	IFF	094	58	FIX
046	00	00	095	42	STO
047	33	X²	096	04	04
048	86	STF	097	92	RTN

098	76	LBL	151	16	16
099	11	A	152	43	RCL
100	86	STF	153	04	04
101	01	01	154	48	EXC
102	19	D'	155	06	06
103	76	LBL	156	67	EQ
104	12	B	157	02	02
105	86	STF	158	25	25
106	02	02	159	32	X:T
107	19	D'	160	66	PAU
108	76	LBL	161	66	PAU
109	13	C	162	76	LBL
110	86	STF	163	33	X²
111	03	03	164	81	RST
112	76	LBL	165	55	÷
113	19	D'	166	05	5
114	91	R/S	167	85	+
115	71	SBR	168	01	1
116	00	00	169	95	=
117	57	57	170	59	INT
118	43	RCL	171	44	SUM
119	03	03	172	00	00
120	42	STO	173	43	RCL
121	08	08	174	01	01
122	32	X:T	175	22	INV
123	43	RCL	176	67	EQ
124	02	02	177	01	01
125	67	EQ	178	82	82
126	01	01	179	04	4
127	65	65	180	44	SUM
128	43	RCL	181	00	00
129	04	04	182	17	B'
130	55	÷	183	43	RCL
131	01	1	184	05	05
132	00	0	185	29	CP
133	09	9	186	67	EQ
134	95	=	187	33	X²
135	42	STO	188	03	3
136	08	08	189	44	SUM
137	22	INV	190	00	00
138	59	INT	191	05	5
139	29	CP	192	32	X:T
140	67	EQ	193	43	RCL
141	02	02	194	08	08
142	01	01	195	22	INV
143	43	RCL	196	67	EQ
144	04	04	197	33	X²
145	32	X:T	198	44	SUM
146	01	1	199	05	05
147	02	2	200	81	RST
148	03	3	201	43	RCL
149	67	EQ	202	08	08
150	02	02	203	85	+

204	01	1
205	04	4
206	95	=
207	55	÷
208	03	3
209	95	=
210	59	INT
211	44	SUM
212	00	00
213	61	GTO
214	01	01
215	82	82
216	01	1
217	01	1
218	44	SUM
219	05	05
220	05	5
221	44	SUM
222	00	00
223	17	B'
224	81	RST
225	02	2
226	01	1
227	61	GTO
228	02	02
229	18	18
230	76	LBL
231	16	A'
232	25	CLR
233	48	EXC
234	00	00
235	22	INV
236	58	FIX
237	91	R/S
238	76	LBL
239	15	E
240	44	SUM
241	00	00
242	43	RCL
243	00	00
244	94	+/-
245	91	R/S
246	42	STO
247	09	09
248	81	RST
249	76	LBL
250	10	E'
251	43	RCL
252	06	06
253	66	PAU
254	66	PAU
255	43	RCL
256	04	04
257	91	R/S

Exemple d'exécution

Touches à presser	Affichage	Signification
5 E	−5	5 jeux disponibles
5 √R/S	−5	idem
D	784	combinaison initiale
B	784	j'arrête la roue du milieu
R/S	488	gagné !
	−6	6 jeux disponibles
D	788	combinaison initiale
B	788	j'arrête la roue du milieu
C	788	j'arrête la roue de droite
R/S	988	gagné !
	−7	7 jeux disponibles
D	755	combinaison initiale
B	755	j'arrête la roue du milieu
C	755	j'arrête la roue de droite
R/S	555	gagné... (encore !)
	−12	12 jeux disponibles

J'ai eu beaucoup de chance. C'est même suspect. Maintenant, continuez la partie : vous avez un crédit de 12 jeux.

Liste des combinaisons gagnantes

Combinaisons gagnantes	Gain	Gain supplémentaire (*)
109	5	3
218	5	3
327	5	3
436	6	3
545	6	3
654	6	3
763	7	3
872	7	3
981	7	3
.00	1	3
.11	1	3
.22	1	3
.33	1	3
.44	1	3
.55	2	3
.66	2	3
.77	2	3
.88	2	3
.99	2	3
000	5	3
111	5	3
222	5	3
333	5	3
444	5	3
555	6	3
666	6	3
777	6	3
888	6	3
999	6	3
123	5	0
dernière combinaison perdante	5	0

(*) Après tirage de 123 ou de la dernière combinaison perdante.

s'immobiliser sur une nouvelle combinaison aléatoire, gagnante ou perdante.

Mais la chance peut aussi vous sourire d'une autre façon. Si vous obtenez la dernière combinaison perdante, ou si vous tirez le nombre 123, le jeu change de nature et devient plus «payant». En effet, pendant 20 tours d'affilée (cas de la dernière combinaison perdante) ou 10 tours (tirage de 123), vos gains en cas de réussite sont augmentés de 3 points et les combinaisons 545, x55 ou 555 vous permettent de jouer 5 tours supplémentaires dans ce mode spécial.

L'affichage se fait alors sous la forme −N.XXX ; N représente votre score et XXX le nombre de tours qu'il vous reste à jouer dans ce mode. Dès que ce dernier nombre est égal à zéro, la partie reprend son cours normal.

Pour obtenir à l'affichage la dernière combinaison perdante (on ne l'a pas toujours en tête), il suffit de presser sur 2nd E'.

Sur les TI-58 et 59, voilà comment se déroule le jeu :

1. Introduisez en E le nombre de parties que vous vous offrez (15 E, par exemple). La machine affiche −N, dans notre cas −15.
2. Entrez un nombre quelconque mais positif et pressez sur R/S. La machine affiche une nouvelle fois −N.
3. Lancez les trois roues en pressant sur D et découvrez la combinaison initiale XYZ. Le ou les zéros situés à la droite du nombre XYZ ne sont pas affichés.
4. Consultez la liste des combinaisons gagnantes et, si vous le voulez, demandez à voir quelle était la dernière combinaison perdante (2nd E').
5. Arrêtez au moins une roue en pressant respectivement sur A, B ou C pour la roue de gauche, celle du milieu ou celle de droite (chiffre des centaines, des dizaines ou des unités). Vous pouvez, bien entendu, immobiliser deux ou trois roues.
6. Remettez la machine en marche grâce à la touche R/S.
7. Si la combinaison finale clignote un court instant, c'est perdu pour cette fois-ci. Si le clignotement persiste un peu, félicitations, vous avez gagné.
8. Enfin, vous retrouvez à l'affichage le nombre de parties que vous pouvez encore jouer (chaque partie valant un franc, vous pouvez vérifier vos gains ou vos pertes).
9. Si vous avez envie de continuer, retournez en 3, sauf si le compteur est à zéro, car dans ce cas, vous devez retourner en 1.

Pour quitter le jeu, pressez sur 2nd A' et admirez à l'affichage la somme que vous auriez gagné si vous aviez dépensé autre chose que des francs parfaitement symboliques.

☐ Pierre Flener

Des mots, encore des mots pour FX-702 P

■ Grâce à des programmes de « logotronique »(1), les ordinateurs de poche deviennent créatifs : ils trouvent des mots nouveaux, originaux et auxquels il peut être amusant de donner des définitions. Ils « inventent » même des mots qui existent !

Le PC-1251 (dans *l'Op* 13 page 35) s'était montré capable d'une telle créativité. Il profitait des instructions DATA et READ présentes dans son Basic.

Le FX-702 P ne possède pas ces deux instructions. Il contourne leur

(1) Le premier programme de « logotronique » de Jean-Pierre Petit et Marc Arondel a été publié par notre confrère L'Ordinateur Individuel, *n° 18 page 53.*

Un pot commun pour toutes les machines

LIST
vous donne rendez-vous
fin juin
chez votre marchand de journaux

Encore des mots
Programme pour FX-702 P
Auteur Thierry Akouka
Copyright *l'Ordinateur de poche* et l'auteur

```
P0: 1579 PAS

1 WAIT 0:PRT "CRE
  ATION DE MOTS":
  GOTO 6
2 A$=MID(H,4):RET

3 A$=MID(I,5):RET

4 A$=MID(J,6):RET

5 A$=MID(K,7):RET

6 P=10*INT (RAN#*
  21)+10:GSB #9:B
  =220:GOTO P
10 $="XYLO BIO DYS
   GEO NEO PAN SU
   B":GSB 2:GOTO B
20 $="AERODAGROAMBI
   ANTICACOCINECRY
   O":GSB 2:GOTO B
30 $="DECIHEMIECTO
   ERGOGONOGYNOHOL
   O":GSB 2:GOTO B
40 $="HOMOHYPOIDEO
   LIPOLOGOLOXOMET
   A":GSB 2:GOTO B
50 $="MISONOSOOCTO
   OLEOOMNIPARAPED
   O":GSB 2:GOTO B
60 $="PERIPOLYPOST
   SADOSEMITHEOXEN
   O":GSB 2:GOTO B
70 $="ANDROASTROCA
   LLICOPROCOSMODE
   RMO":GSB 3:GOTO
    B
80 $="DROMOGONIOGY
   NNOHECTOHELIONI
   ERO":GSB 3:GOTO
    B
90 $="HIPPONOMEOHY
   DROHYPERICONOIN
   FRA":GSB 3:GOTO
    B
100 $="MACRONETEOMI
    CRONNEMOMYTHONE
    CRO":GSB 3:GOTO
     B
110 $="ORTHOPALEOPA
      THOPENTAPETROPH
      AGO":GSB 3:GOTO
       B
120 $="PHILOPHOBOPH
      ONOPHOTOPORNOPR
      OTO":GSB 3:GOTO
       B
130 $="RADIORETRORH
      INOSTENOTRACHYTE
      LEO":GSB 3:GOTO
       B
140 $="BIBLIOCLEPTO
      CHIMIOCHRONOCRY
      PTO":GSB 4:GOTO
       B
150 $="DEXTRODODECA
      DYNAMOEMBRYOENT
      OMO":GSB 4:GOTO
       B
160 $="GASTROGLOSSO
      HEMATOHETEROLAT
      ERO":GSB 4:GOTO
       B
170 $="MEGALOMORPHO
      NUCLEOPAPYROPHA
      LLO":GSB 4:GOTO
       B
180 $="PHRENOPHYSIO
      PSEUDOPSYCHOSTE
      REO":GSB 4:GOTO
       B
190 $="SCLEROSCHIZO
      SIMILISPELEOTEC
      HNO":GSB 4:GOTO
       B
200 $="ELECTROEROTI
      COPHANERO THERM
      O":GSB 5:GOTO B
210 $="  TRIBO  ULT
      RA  VIDEO    ZO
      O":GSB 5:GOTO B
220 GSB #9:Q=10*INT
      (RAN#*11)+240:
      D=350:GOTO Q
230 C$=MID(H,4):RET

231 C$=MID(I,5):RET

232 C$=MID(J,6):RET

233 C$=MID(K,7):RET

240 $="DOXEDYNEFUGE
      GAMEGENEGYNEMAN
      E":GSB 230:GOTO
       D
250 $="ALGIECARDECY
      CLEDERMEDROMEGR
      ADE":GSB 231:GO
      TO D
260 $="LATRELOGIELO
      GUEMANIENAUTENO
      MIE":GSB 231:GO
      TO D
270 $="PATHEPHAGEPH
      ILEPHOBEPHONERR
      HEE":GSB 231:GO
      TO D
280 $="CINESECLASTE
      COSMOSGASTREGLO
      TTE":GSB 232:GO
      TO D
290 $="GENESELATERE
      MANCIEMETRIEMNE
      SIE":GSB 232:GO
      TO D
300 $="MORPHEPHANIE
      SCAPHESCOPIEZOA
      IRE":GSB 232:GO
      TO D
310 $="CEPHALEDACTY
      LEDIDACTEMATIQU
      E":GSB 233:GOTO
       D
320 $="MEGALIENEVRO
      SESTHENIETHEISM
      E":GSB 233:GOTO
       D
330 $="TONIQUETYPE
       VORE   TRON
      ":GSB 233:GOTO
       D
340 $="TROPE  STASE
       PNEE   PYGE
      ":GSB 233:GOTO
       D
350 PRT A$+C$:GOTO
    6

P9: 74 PAS

10 R=INT (RAN#*7):
   H=1+R*4:S=INT (
   RAN#*6):I=1+S*5
   :T=INT (RAN#*5)
20 J=1+6*T:U=INT (
   RAN#*4):K=1+7*U
   :RET
```

absence par un programme long mais un principe simple.

Des préfixes et des suffixes sont stockés dans la variable $ (lignes 10 à 210 et 240 à 340). Pour extraire un préfixe, le programme commence par tirer un numéro de ligne au hasard, entre 10 et 210 (ligne 6 de P0). Selon la ligne tirée, le préfixe aura 4 lettres (lignes 10 à 60), 5 lettres (lignes 70 à 130), 6 lettres (lignes 140 à 190) ou 7 lettres (lignes 200 et 210).

Alors, il ne reste qu'à tirer le nombre à partir duquel 4, 5, 6 ou 7 lettres seront prises.

Si, par exemple, c'est la ligne 20 qui a été tirée, les préfixes auront une longueur de quatre lettres et le programme renverra à la ligne 2. Là, l'instruction utilisée sera MID (H, 4) où H aura été calculé à la ligne 10 de P 9 : H = 1+R*4, R étant un nombre aléatoire compris entre 0 et 6. Ainsi, le préfixe retenu sera stocké dans A$. Le même principe sert à la recherche du suffixe qui sera placé dans C$.

Le programme n'a plus qu'une chose à faire : imprimer A$ + C$. Un nouveau mot est né. S'il ne vous plaît pas, attendez un peu pour en voir un autre. Le FX-702 P vous en réserve plusieurs milliers...

Exemple d'exécution

```
CREATION DE MOTS
CRYPTOPHILE
 DYSMORPHE
ECTODOXE
STEREOGENESE
PHILONOMIE
PROTOPYGE
PHANERODIDACTE
COSMOPHEE
  VIDEORRHEE
CHRONOMNESIE
PHANERONEVROSE
MICROTROPE
PHOBOMATIQUE
PERIPHILE
DYNAMOPHONE
PSYCHORRHEE
PHAGOMANE
DROMOMORPHE
MYTHORRHEE
```

☐ Thierry Akouka

ORDI-5

le magazine de votre

SINCLAIR

SPECTRUM
et ZX-81

Si vous utilisez un ordinateur SINCLAIR (ZX 81*, ZX 80* ou Spectrum*) ou si vous comptez en acheter un, sachez que la revue **ORDI-5** a été créée pour vous. Indépendante de tout constructeur ou importateur, **ORDI-5** vous fournit quatre fois par an des programmes, des conseils, des astuces, de nouvelles idées d'utilisation. **ORDI-5** teste pour vous en toute objectivité et indépendance les produits matériels et logiciels adaptables sur votre SINCLAIR. **ORDI-5** vous tient au courant de toutes les nouveautés susceptibles de vous intéresser. Commandez un numéro ou... **abonnez-vous**, vous économiserez 20 %.

*marques déposées

ORDI-5, pour tirer bien plus de votre SINCLAIR

BON DE COMMANDE

à retourner à ORDI-5, 8 rue Saint-Marc 75002 PARIS

Nom _____

Adresse _____

Pays _____ Code postal _____ Ville _____

☐ Je désire recevoir les 4 derniers numéros parus et m'abonner pour recevoir les 4 prochains numéros. (France 160 FF; Etranger** 180 FF; par avion 320 FF).

☐ Je désire recevoir les numéros antérieurs suivants : _____ (prix d'un n° 25 FF; Etranger** 30 FF; par avion 40 FF).

☐ Je désire m'abonner à ORDI-5 pour 1 an, 4 n°s à partir du n° _____ (tarif France 80 FF; Etranger** 90 FF; par avion 160 FF). (Actuellement ORDI-5 est trimestriel).

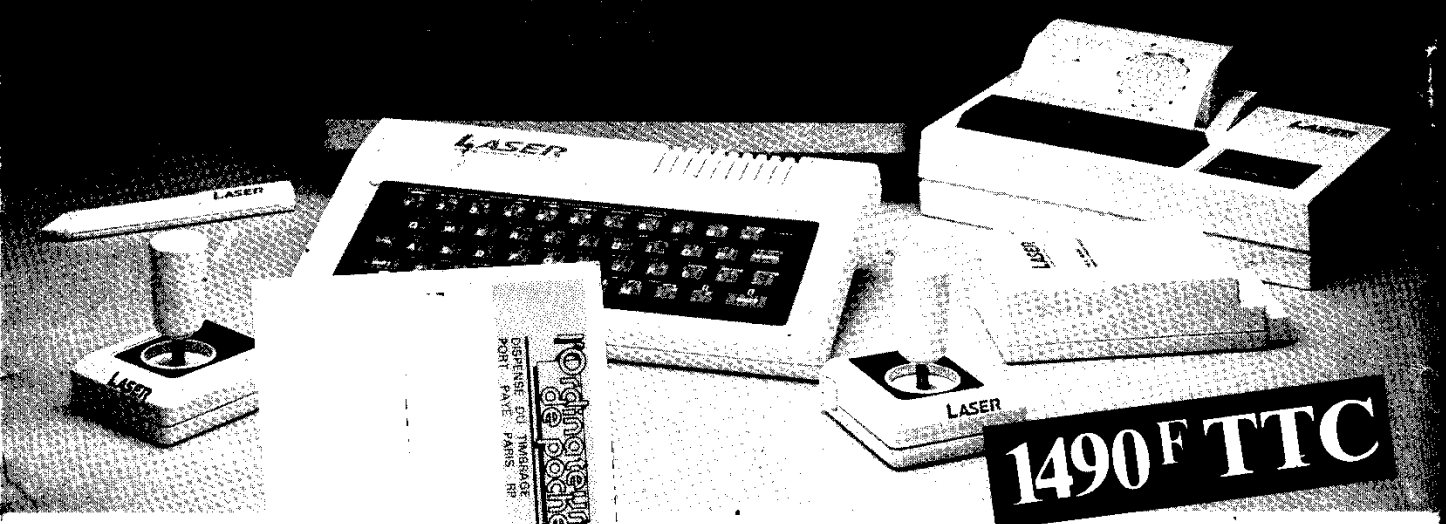
Ci-joint mon règlement indispensable par chèque bancaire ☐ chèque postal ☐ virement ☐

**Pour les pays autres que la France, utiliser un virement en FF compte Crédit Lyonnais Paris n° 30002 00402 8455 J. Les frais de virement sont à la charge de l'acheteur.

LE LASER 200

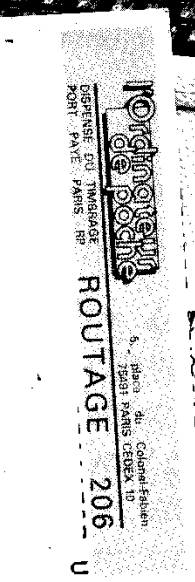
UN MICRO ORDINATEUR COULEUR SECAM

VRAIMENT TRÈS ÉTONNANT.



1490 F TTC

Micropro ngage Microsoft Basic • Affichage direct
antenne tél r 45 touches pleine écriture, + clef d'entrée,
+ graphism i-erreurs... • Texte + graphismes mixables
9 coul rrection plein écran • Son incorporé
• ension + 16 K + 64 K,
interface im te,
stylo optique, manettes,
jeux, modem,
disquettes...



BON DE COMMANDE

A retourner à : VIDEO TECHNOLOGIE - 19, rue Luisant - 91310 Montlhéry
Tél. (6)901.93.40 - Télex SIGMA 180114

OP

Je désire recevoir :
LASER 200 SECAM comprenant :
Le LASER 200 avec son modulateur SECAM incorporé se branchant directement sur l'antenne du téléviseur.
+ Câble de liaison fiches jack pour lecteur de K7
+ Câble de liaison micro/télé ou moniteur
+ Livre technique (150 pages) de BASIC
+ Livret d'exercices
+ Manuel de mise en route
+ Cassette de démonstration en français
+ Garantie ..... 1.490 F TTC

EXTENSION-PERIPHERIQUES-INTERFACES LASER 200
Extension mémoire 16K ..... 590 F TTC
Extension mémoire 64K ..... 1.190 F TTC
Lecteur préréglé de cassettes type DR 10 ..... 570 F TTC
Paire de manettes de jeux avec son interface ..... 320 F TTC
Interface d'imprimante "Centronic parallele" ..... 320 F TTC
Imprimante 4 couleurs papier standard ..... 2.190 F TTC
Interface disquette ..... (en préparation) ... N.C.
Stylo optique ..... (en préparation) ..... N.C.

LOGICIELS LASER 200
Cassettes avec programmes 4K ou 16K ... 79 F TTC
(Voir liste détaillée constamment augmentée)

TOTAL DE MA COMMANDE :

Je choisis de payer le total de ma commande :
☐ Au comptant, par CCP, chèque bancaire, ou mandat, à l'ordre de VIDEO TECHNOLOGIE FRANCE
☐ Contre-remboursement au transporteur, moyennant une taxe de 60 F.

Nom
Prénom
N°
Rue
Ville
Code Postal

Signature

Liste de plus de 100 revendeurs, sur simple demande

VIDEO TECHNOLOGIE FRANCE

19, rue Luisant - 91310 Montlhéry
Tél. (6)901.93.40
Télex SIGMA 180114